Machine learning Calabi-Yau metrics: an energy functional approach

Lucille Calmon

Project supervised by Prof Yang-Hui He

September 25, 2020

Submitted in partial fulfilment of the requirements for the degree of Master of Science of Imperial College London

Abstract

In this thesis, we consider numerical Calabi-Yau metrics. We present an overview of currently available methods and we introduce the reader to the energy functional approach developed by Headrick and Nassar (1). Their method is compared to Donaldson algorithm (2; 3). We describe recent work (4) presenting a hybrid approach that uses machine learning to significantly speed up Donaldson's method. Inspired by (4) we propose to replace the minimisation step in (1) by a simple neural network, hoping to provide a faster version of (1). Using supervised learning, we demonstrate that our network can predict the Kähler potential in a matter of seconds on the Calabi-Yau Quintic with 1.5% error after 2 minutes of training. We establish the robustness of our predictions and ponder whether our alternative method could be viably used for computing Calabi-Yau metrics on threefolds beyond the Quintic. We offer further directions of research to consolidate our results and answer remaining questions.

Contents

1 String theory, compactification and machine learning

String theory, a potential quantum theory of fundamental forces including gravity, describes particles as extended objects, so-called "strings" instead of points. Quantisation of the string is subtle, yet the particle spectrum can be obtained. It turns out, however, that the spectra of the quantised theories are at odds with observable physics. Even worst, bosonic string theory contain tachyons, i.e. particles with negative rest mass going faster than the speed of light. This goes against the fundamental axioms of relativity. Bosnic string theory additionally can only be consistently written in $d = 26$ spacetime dimensions.

Physical features predicted by the theory are however promising and all forces including gravity can be found in the spectrum. To the best of our knowledge, the spacetime we live in is 4 dimensional and contains both bosonic and fermionic excitations. The latter can be introduced by adding "supersymmetry" on the string's worldsheet. In supersymmetric string theories (referred to as superstring theories), all known fundamental fermionic and bosonic particles in the Standard Model are paired with a "super-partner" of respectively bosonic and fermionic nature. The introduced symmetry can also remove the tachyon from the spectrum, leaving a consistent theory. In the world we see, there doesn't seem to be any hints of SUSY. A possible resolution is the following: at the low energy scales that we experience, supersymmetry could be broken. The super-partners acquire mass above the energy scales we probe, thus explaining why they are not observable in collider experiments.

The resulting superstring theories (type I, IIA, IIB and the two heterotic ones) then need to be formulated in "only" 10 dimensional spacetime to be consistent. Assuming the observable universe consists of 4 spacetime dimensions, there remain 6 auxiliary dimensions. These are understood to be "compactified", in contrast with the non-compact directions spanned by the 4 spacetime dimensions. These redundant dimensions can be pictured as "curled up" around a space of negligible scale compared to the world we experience. Consequently, they become are imperceptible at the scales we can probe in experiments.

Mathematically, the spacetime is a 10 dimensional manifold formulated within the framework of differential geometry. This space is then decomposed into the product of a $4d$ space e.g. Minkowski, and a $6d$ compact space (that is described using a 3 dimensional complex manifold, also called threefold). The geometrical characteristics of the compactification space (called internal space) determine the properties of the physics observed in 4 dimensions. Observed 4d physics then restricts the possible choices of internal spaces. In 1985, Candelas et al. (5) provided the first description of the properties and conditions that the compactification threefolds must obey in order to predict Standard Model (SM) like physics in Minkowski space. In practice, the resulting effective theories remain supersymmetric. With the correct internal space and boundary conditions they can describe so-called Minimally Supersymmetric Standard Models (MSSM) which look like the SM at low energy scales.

Additionally, the 10-dimensional manifold and consequently the compactification threefold, need to obey the vacuum Einstein equation. In particular,the Ricci-tensor associated to the metric on the internal space must vanish. Such a metric is called

Ricci-flat. It was also found in (5) that the compactification spaces must also be Kähler manifolds. The natural candidates for the internal space are then Ricci-flat Kähler manifolds, named Calabi-Yau manifolds in (5) after the academics who studied them.

The characteristics of the compactification space can be restricted further by imposing that the MSSM admits the SM gauge group and contains 3 generations of quarks and leptons. The following was initially derived in (5) (see (3) for an accessible review). In practice, the original string theory admits a larger gauge group (unified group) that contains the SM group. In order to break the unified group to the correct correct SU(3)×SU(2)×U(1) SM gauge group at low energies, the compactification space must admit a Yang-Mills connection such that the internal gauge group commutes with the SM gauge group inside the unified group. Additionally, the observed 3 generations of leptons and quarks then restricts the topology of the space, imposing $\chi = \pm 6$, where χ is the Euler characteristic of the Calabi-Yau.

Soon after these conditions were derived in (5), examples of appropriate Calabi-Yaus where constructed in (6). These were then used to construct superstring theory models in (7; 8). Since then, about 10^{10} Calabi-Yau threefolds have been constructed and classified in the monumental Kreuzer-Skarke (KS) data set (9). Of the 10^{10} constructed spaces, about 10^6 admit $\chi = \pm 6$ (10). In fact, while the number of Calabi-Yau threefolds was conjectured to be finite (11), it is still unclear how many there could be.

Any of these threefolds could be used as a compactification space. This forms

the "Calabi-Yau landscape". Every threefold in the landscape then leads to different physics on Minkowski space. However, beyond the choice of Calabi-Yau, choices of boundary conditions and flux data also affect the physics observed and content of the theory. This is the "string landscape", which can contain up to 10^{500} possible configurations in the case of type IIB string theory (12). This provides an interesting playground at the boundary of mathematics, theoretical physics and data science. This exciting field of research is elegantly introduced in (10).

The question now is, while each Calabi-Yau manifold in the landscape corresponds to a universe, which one could result in *our* universe? Unfortunately, this is an arduous question to answer. It decomposes into two parts: how can we predict the physical properties of the physics associated to a given Calabi-Yau and how can we find the correct one(s) in the numerous possibilities. The latter could be tackled with machine learning. This line of research was simultaneously imagined by different collaborations in 2017 (13; 14; 15; 16). Following a paradigm imagined by Yang-Hui He (17) to probe the landscape with machine learning techniques, libraries of all Calabi-Yau spaces with some of their topological properties (like the KS set from (9)) have been analysed to spot patterns and discover potential compactification spaces. More details can be found in (18; 10).

Predicting physical observables from string theory to compare them to the SM measured parameters is a more complicated story. While gauge and Higgs bosons masses, calculated in (19) for (20), are found to be compatible with observations at low energies, the masses of the fundamental fermions have never been computed. They can in principle be determined from cubic Yukawa couplings that

depend in particular on the geometry and Yang-Mills connection of the Calabi-Yau. The couplings have been calculated using topological data (21) but this yields non-normalised values that can't be compared against the known measured parameters. Obtaining canonically normalised couplings requires, in particular, knowledge of the Ricci-flat metric and Yang-Mills connection coefficients on the tangent bundle of the compactification space (see the introduction of (4) and (22) for details).

Unfortunately, the metric and connection coefficients are particularly arduous to calculate. While Calabi-Yau manifolds can be easily constructed as described in (10), there exists, as of now, no known closed form for the Ricci-flat metric on any non trivial Calabi-Yau. While analytic metrics could exist, finding them is classified as a "hard problem" by (22). Consequently, efforts have been deployed to compute the metrics numerically (23; 3; 24; 1).[1] We now present some of these numerical methods.

The first Ricci-flat metrics were calculated in 2005 in (23). They introduced a lattice and coordinate patches on a family of Calabi-Yau twofolds and solved the Einstein equations on the lattice. This yielded metrics within 1% accuracy in a few days of computation. The extent of the lattice used was limited by the computation time, while memory requirements kept the method from being applicable to most threefolds.

Shortly after, a different method was developed in (3; 25) using theoretical work by Donaldson (2). The idea is to expand the metric in a basis of complete func-

[1] We note that such a method would generalise to the connection coefficients.

tions (akin to Fourier modes) and find appropriate expansion coefficients. This "momentum representation" of the metrics was a significant improvement as storing coefficients and avoiding patches on the Calabi-Yau requires less computational resources than (23). The method itself consists in applying a map iteratively to the coefficients. This was shown to converge to a so-called balanced metric, which is an approximation to a Ricci-flat metric. This produced metrics of similar accuracy to the ones found in (23) through an easier implementation and shorter computation time (order of a few hours to a few days depending on the accuracy).

Additionally, using the same expansion of the metric, (1) turned the question into a minimisation problem by defining an energy functional that has a unique minimum on the Ricci-flat metric. This method has the advantage of being computationally inexpensive and mathematically elegant while yielding results comparable in accuracy to its predecessors. The metrics can be generated in times of the order of a few minutes to a few hours and the accuracy is found to improve more quickly than Donaldson's approach. The set up and results will be explained in detail in section 3.2 and forms the basis of our work.

These methods were all successfully developed to produce metrics of a sufficient accuracy on highly symmetric Calabi-Yaus (in particular the so-called Fermat Quintic), thus acting initially as proofs of principle. Donaldson's method has since been developed further and used to compute connection coefficients (26) as well as to predict spectra of the scalar Laplace operator (27).

Following up from (3), a hybrid method incorporating machine learning elements

was found to speed up the original method by a factor of 50 with little loss in accuracy (4). We describe the work of (4) in more details in section 3.1. This approach is currently investigated further to compute metrics on the Calabi-Yau used in (25).

Inspired by the work in (4), which mixes machine learning and Donaldson algorithm, we apply and present a similar strategy to the promising energy functional method (1).[2] We aim to replace the ultimate minimisation of the functional by a simple neural-network in order to speed up the computation of the metrics. In such a hybrid approach, the data required for the minimisation of the functional becomes the input of the neural network where we use supervised learning to predict the result of the minimisation. We apply this to the Quintic Calabi-Yau and hope to pave the way for an alternative viable method that can yield results for more sophisticated Calabi-Yau manifolds and contribute to finding a unified description of our universe through string theory.

This thesis is organised as follows. The first section introduces elements of complex geometry – starting from complex manifolds, the complexification of the tangent space and complex forms – then introduces Hermitian manifolds and Kähler geometry. This lays the foundation required to define Calabi-Yau manifolds in section 2.6. We briefly discuss different definitions and introduce complex projective space and the Calabi-Yau Quintic in section 2.7. In section 3, we introduce Donaldson algorithm briefly and show how the use of machine learning significantly improves the method. We then discuss in section 3.2 the energy functional method. We present all the elements required to construct the functional and discuss the algorithm. In

[2]We refer to the method of (1) as the "Headrick and Nassar method" or the "Energy functional method" throughout this thesis

section 4, we introduce elements of machine learning and present our results, building up from the energy functional method. Finally, we discuss the significance of our results and conclude in section 5.

2 Complex geometry and Calabi-Yau manifolds

In the following section, we reproduce definitions (from Chapter 8 in (28)) on a "need to know" basis where we have selected necessary concepts required to define Calabi-Yau manifolds and the energy functional method. We extend the definition of a real manifold to that of a complex manifold, introduce the complexified tangent space and complex differential forms and finish by defining Hermitian and Kähler manifolds. We finally define Calabi-Yau manifolds and discuss some of their properties.

We assume the reader is familiar with real differentiable manifolds and the differential geometry jargon. We refer the reader to the QFFF course notes for Differential Geometry (29) and the book "Geometry, Topology and Physics" by Nakahara (28) (especially chapters 5, 6 and 8) for an introduction to differential geometry.

About the wording of definitions: the author has learned content of this section from the book "Geometry and topology for physics" by Nakahara. An effort is made to independently phrase definitions given in this thesis in the clearest and most concise way and this sometimes simply coincides with the wording from (28). The choice was made not to overcomplexify the definitions and we acknowledge that some of the definitions in this section may be written coincidentally in the exact or very similar form as given initially by Nakahara in (28).

2.1 Complex manifolds and the complex structure

Loosely speaking, an m-dim real ($\dim_{\mathbb{R}} = m$) differentiable manifold is a smooth space that is locally isomorphic to $\mathbb{R}^m$. In analogy, an m-dim complex ($\dim_{\mathbb{C}} = m$) manifold is bi-holomorphic to $\mathbb{C}^m$. More formally,

Definition 1. *A real differentiable manifold is a Hausdorff topological space* $(\mathcal{M}, \mathcal{O})$.[3] $\mathcal{M}$ *has an atlas* (U_i, ψ_i) *such that:*

1. *the charts* $U_i \in \{\mathcal{O}\}$ *cover* $\mathcal{M}$*:* $\bigcup_i U_i = \mathcal{M}$
2. ψ_i *are homeomorphisms into a subset of* $\mathbb{R}^m$
3. *in any overlap* $U_i \cap U_j \neq \varnothing$*, the maps* $\phi_{ij} = \psi_i \circ \psi_j^{-1}$ *(called transition functions) are infinitely differentiable.*

The homeomorphisms ψ_i locally (and smoothly) map all points in the manifold onto $\mathbb{R}^m$. The transition functions specify the particular structure of $\mathcal{M}$. The image of ψ_i in $\mathbb{R}^m$ is called a coordinate patch and differential calculus can be applied as usual there. The smoothness of the transition functions allows this to be extended from patch to patch and to the entirety of $\mathcal{M}$.

Two types of objects can be defined on $\mathcal{M}$: global and local structures. A local structure is defined at a point $p \in \mathcal{M}$ or in a neighborhood of p but cannot be smoothly glued over the whole manifold. A global structure is defined everywhere on $\mathcal{M}$ and transforms smoothly between coordinate patches. Most of the structures following are defined at a point (highlighted throughout by the use of the subscript p) and naturally extend to the whole manifold to form a field. For example, a tensor

[3] We label the topology of $\mathcal{M}$ by the collection of open sets $\{\mathcal{O}\}$ and keep the conventional J for the complex structure.

field, where a tensor is smoothly associated to each point on $\mathcal{M}$ is an example of a global structure.

A complex manifold is defined analogously to definition 1 with the requirement that the homeomorphisms ψ_i map open sets of $\mathcal{M}$ to $\mathbb{C}^m$ and transition functions ϕ_{ij} are holomorphic on any overlap. Formally,

Definition 2. *An m-dim complex manifold is a Hausdorff topological space $(\mathcal{M}, \prime)$. $\mathcal{M}$ has an atlas (U_i, ψ_i) such that:*

1. *the charts $U_i \in \{\mathcal{O}\}$ cover $\mathcal{M}$: $\bigcup_i U_i = \mathcal{M}$*
2. *ψ_i are homeomorphisms from U_i into a subset of $\mathbb{C}^m$*
3. *in any overlap $U_i \cap U_j \neq \varnothing$, the maps $\phi_{ij} = \psi_i \circ \psi_j^{-1}$ are holomorphic.*

Since $\mathbb{C} \simeq \mathbb{R}^2$, in a coordinate patch, an m-dim complex manifold (with m complex coordinates z^μ, $\mu = 1, 2, 3...m$) is locally equivalent to a $2m$-dim real manifold (with $2m$ real coordinates $\{x^\mu, y^\mu\}$) under the correspondence $z^\mu = x^\mu + iy^\mu$.[4] The holomorphicity of the transition functions between patches guarantees that this equivalence holds for the entire manifold. This is true of any m-dim complex manifold and allows us to use the structures and calculus originally developed on real manifolds on complex manifolds by viewing them as a $2m$ real manifold. While both are the same manifold, we use $\mathcal{M}_\mathbb{R}$ when considering the $2m$-dim associated real manifold and reserve $\mathcal{M}$ to the complex manifold viewed as complex.

Clearly, an odd-dim real manifold cannot be complex and even-dimensional real manifolds aren't in general complex manifolds. Complex manifolds instead come

[4]We use m for both complex and real dimensions depending on context.

equipped with a "complex structure" related to the additional requirement imposed on transition functions.

Definition 3. *The complex structure of a complex manifold $\mathcal{M}$ is a linear map $J_p : T_p\mathcal{M}_{\mathbb{R}} \longrightarrow T_p\mathcal{M}_{\mathbb{R}}$. This endomorphism of the tangent space obeys:*

1. *it squares to $-\mathrm{id}_{\mathrm{T_p}\mathcal{M}_{\mathbb{R}}}$,*
2. *it can be put in the form $J(\frac{\partial}{\partial x^\mu}) = \frac{\partial}{\partial y^\mu}$; $J(\frac{\partial}{\partial y^\mu}) = -\frac{\partial}{\partial x^\mu}$ at a given point $p \in \mathcal{M}$*
3. *it can be defined globally on $\mathcal{M}$, in which case it is labelled J.*

While any real manifold admits a tensor that locally squares to $-\mathrm{id}_{\mathrm{T_p}\mathcal{M}_{\mathbb{R}}}$, such a structure can only be globally defined on complex manifolds. This object thus fully encodes the complex structure of $\mathcal{M}$.

2.2 Complexification and tangent spaces

Complex manifolds also admit tangent and co-tangent spaces which are derived from the complexification of the tangent and cotangent spaces of the equivalent $2m$-real manifold ($\mathcal{M}_{\mathbb{R}}$).

We start by describing how objects on $\mathcal{M}$ can be extended from $\mathcal{M}_{\mathbb{R}}$ to $\mathcal{M}$. Generally: real linear operators are said to be "extended" to act on complex domains and vector spaces are "complexified" once their elements are extended.

We start with smooth functions and remind the reader that a smooth function on $\mathcal{M}_{\mathbb{R}}$, $f : \mathcal{M}_{\mathbb{R}} \longrightarrow \mathbb{R}$, locally maps a point in $\mathcal{M}_{\mathbb{R}}$ to $f(x, y) \in \mathbb{R}$. The set of all such functions is denoted $\mathcal{F}(\mathcal{M})$.

Such a real smooth function can be defined on $\mathcal{M}$ in complex coordinates by rewriting $\{x^\mu, y^\mu\}$ in terms of $\{z^\mu, \bar{z}^\mu\}$, seen as coordinates on $\mathcal{M}$. This allows us to define the set of complex functions and its elements as follows.

Definition 4. *The complexified space of smooth functions is the set*

$$\mathcal{F}(\mathcal{M})^{\mathbb{C}} = \{h = f + ig : \forall f, g \in \mathcal{F}(\mathcal{M})\}. \tag{1}$$

Elements of $\mathcal{F}(\mathcal{M})^{\mathbb{C}}$ are complex functions on $\mathcal{M}$. In general, any complex function $h(z, \bar{z})$ on $\mathcal{M}$ can then be written in the form $h(z, \bar{z}) = f(z, \bar{z}) + ig(z, \bar{z})$ where $f, g \in \mathcal{F}(\mathcal{M})$ are real functions.

Definition 5. *The conjugate of a complex function h is $\bar{h}(z, \bar{z}) = f(z, \bar{z}) - ig(z, \bar{z})$.*

Definition 6. *A complex function $h \in \mathcal{F}(\mathcal{M})^{\mathbb{C}}$ is real if $h(z, \bar{z}) = \bar{h}(z, \bar{z})$.*

We extend this to vectors and tensors defined at a point, which then generalise to fields over $\mathcal{M}$. A real vector is extended to act on complex functions as follows.

Definition 7. *A real vector, seen as a linear map $A : \mathcal{F}(\mathcal{M}) \longrightarrow \mathbb{R}$ is extended to act on complex valued functions from $\mathcal{F}(\mathcal{M})^{\mathbb{C}}$ and becomes $A(h) = A(f) + iA(g)$ for $h = f + ig \in \mathcal{F}(\mathcal{M})^{\mathbb{C}}$.*

Complex vectors can be defined as follows.

Definition 8. *Given two real vectors X and Y, the vector $Z = X + iY$ is a complex vector. The complex conjugate of Z is $\bar{Z} = X - iY$. A complex vector obeying $Z = \bar{Z}$ is a real vector.*

Definitions 7 and 8 can be extended to vector fields where they now hold at each point in $\mathcal{M}$ smoothly patched together.

Identically to spaces of functions, vector spaces (at a point) are complexified to obtain a complex vector space with double the starting real dimension.

Definition 9. *The complexification $V^{\mathbb{C}}$ of a real vector space V is the set $V^{\mathbb{C}} = \{Z = X + iY : X, Y \in V\}$. $V^{\mathbb{C}}$ is also a vector space with rules for addition and scalar multiplication inherited from the structure of V itself with elements extended to act on complex functions.*

The space V is contained in $V^{\mathbb{C}}$, and the real dimension of V becomes the complex dimension of $V^{\mathbb{C}}$.

As real vectors in definition 7, all linear operators are "extended" to act on complex arguments. For example, a linear operator defined on V, $A : V \longrightarrow \mathbb{R}$, is extended to $V^{\mathbb{C}}$ such that $A : V^{\mathbb{C}} \longrightarrow \mathbb{C}$. The action of A on elements in $V^{\mathbb{C}}$ is determined from linearity such that $A(X + iY) = A(X) + iA(Y)$. This applies to all linear operators on $\mathcal{M}$. In this manner, general tensors are extended (at every point) and spaces of tensors (which are vector spaces) are complexified.

Definition 10. *Given two real tensors t_1 and t_2 at p, both of rank (p, q), the tensor $t = t_1 + it_2$ is a complex tensor at p. The conjugate of t is $\bar{t} = t_1 - it_2$. t is real if $\bar{t} = t$.*

This process straightforwardly generalises to tensor fields allowing us to construct the complexified tangent and co-tangent bundles of $\mathcal{M}$.

The tangent space of $\mathcal{M}_{\mathbb{R}}$ is spanned locally, in coordinate basis, by $\{\partial/\partial x^1, \partial/\partial x^2, ..., \partial/\partial x^m, \partial/\partial y^1, \partial/\partial y^2, ..., \partial/\partial y^m\}$. It is straightforward to show that the complexified tangent space $T_p\mathcal{M}^{\mathbb{C}}$ is spanned by the $2m$ complex basis

vectors

$$\begin{aligned}\frac{\partial}{\partial z^\mu} &= \frac{1}{2}\Big\{\frac{\partial}{\partial x^\mu} - i\frac{\partial}{\partial y^\mu}\Big\},\\ \frac{\partial}{\partial \bar{z}^\mu} &= \frac{1}{2}\Big\{\frac{\partial}{\partial x^\mu} + i\frac{\partial}{\partial y^\mu}\Big\}.\end{aligned} \tag{2}$$

We highlight that this basis is $2m$-complex dimensional and consequently $4m$-real dimensional, double the real dimension of the real tangent space that is complexified. This is a feature of complex manifolds where points are identified with both z and $\bar{z}$ coordinates.

Vectors of the form $V = V^\mu \frac{\partial}{\partial z^\mu}$ and $V = V^\mu \frac{\partial}{\partial \bar{z}^\mu}$ are respectively called holomorphic and anti-holomorphic vectors.[5] Generally, vectors are linear combinations of both types of basis vectors.

Similarly, the complexified co-tangent space is spanned by the basis

$$\begin{aligned}\mathrm{d}z^\mu &= \mathrm{d}x^\mu + i\mathrm{d}y^\mu,\\ \mathrm{d}\bar{z}^\mu &= \mathrm{d}x^\mu - i\mathrm{d}y^\mu,\end{aligned} \tag{3}$$

which again is $2m$-complex dimensional. It can be easily checked that these bases are dual to each other (as one would expect). In particular, they obey

$$\begin{aligned}\Big\langle \frac{\partial}{\partial z^\mu}, \mathrm{d}\bar{z}^\nu \Big\rangle = \Big\langle \frac{\partial}{\partial \bar{z}^\mu}, \mathrm{d}z^\nu \Big\rangle = 0,\\ \Big\langle \frac{\partial}{\partial z^\mu}, \mathrm{d}z^\nu \Big\rangle = \Big\langle \frac{\partial}{\partial \bar{z}^\mu}, \mathrm{d}\bar{z}^\nu \Big\rangle = \delta^\nu_\mu.\end{aligned} \tag{4}$$

These complexified tangent and co-tangent spaces are the tangent and co-tangent spaces of the complex manifold $\mathcal{M}$.

As all linear operators, the complex structure J is then extended to act on

[5] As will be explained in detail in the case of forms, holomorphic and anti-holomorphic vectors are sometimes referred to as (1,0) and (0,1) vectors respectively. Note that the components of holomorphic vectors are not necessarily holomorphic functions of z.

$T_p(\mathcal{M})^{\mathbb{C}}$. In the basis defined in eq. 2, J takes the form

$$J\Big(\frac{\partial}{\partial z^\mu}\Big) = i\frac{\partial}{\partial z^\mu},$$
$$J\Big(\frac{\partial}{\partial \bar{z}^\mu}\Big) = -i\frac{\partial}{\partial \bar{z}^\mu},$$

thus highlighting that holomorphic and anti-holomorphic vectors are invariant spaces of J. This defines two independent subspaces of $T_p(\mathcal{M})^{\mathbb{C}}$: the holomorphic tangent space

$$T_p\mathcal{M}^+ = \{V \in T_p\mathcal{M}^{\mathbb{C}} : J(V) = +i\}$$

and its anti-holomorphic counterpart

$$T_p\mathcal{M}^- = \{V \in T_p\mathcal{M}^{\mathbb{C}} : J(V) = -i\}.$$

The complexified tangent space is the direct sum of $T_p\mathcal{M}^-$ and $T_p\mathcal{M}^+$.

The split between holomorphic and anti-holomorphic tangent spaces is entirely determined by J. This split is preserved by the transition functions and is therefore independent of the coordinates worked with. This is the characteristic feature of complex manifolds which allows significantly more structure to be defined and considered, some of which we introduce in section 2.4.

2.3 Differential forms and differential calculus

We give here the main properties of complex differential forms required to understand the geometry of Calabi-Yau manifolds presented in section 2.6 and the methods developed to compute Ricci-flat metrics detailed in 3. Forms being particular tensors, the following are direct consequences of definitions 9 and 10.

Real differential forms are extended to act on complexified vector spaces, as explained in section 2.2. Complex differential forms are constructed as follows.

Definition 11. *Given two real q-forms at a point p, ω and η, the superposition $\zeta = \omega + i\eta$ is a complex p-form. The conjugate of ζ is $\bar{\zeta} = \omega - i\eta$ and a form obeying $\zeta = \bar{\zeta}$ is a real form.*

The space of real q-forms at p, denoted $\Omega^q_p(\mathcal{M})$ is extended then complexified.

Definition 12. *The space of complex differential forms is defined as*

$$\Omega^q_p(\mathcal{M})^{\mathbb{C}} = \big\{\omega + i\eta : \omega, \eta \in \Omega^q_p(\mathcal{M})\big\}.$$

The space $\Omega^q_p(\mathcal{M})^{\mathbb{C}}$ is a vector space under addition and multiplication by a scalar. This is inherited from the structure of $\Omega^q_p(\mathcal{M})$ and the linearity of the elements.

This generalises straightforwardly to fields where a q-form is smoothly assigned to every point in $\mathcal{M}$. When referring to the space of q-form fields, the subscript p is dropped. q-form fields are extended at every point and the space of all q-form fields is complexified to obtain $\Omega^q(\mathcal{M})^{\mathbb{C}}$, the space of all complex q-forms.

The exterior product and derivative are naturally extended to act on complex forms.

Definition 13. *Given two complex q-forms ζ and ξ with decomposition $\zeta = \omega + i\eta$, $\xi = \alpha + i\beta$ for ω, η, α and β real q forms, the exterior product of ζ and ξ is defined as:*

$$\zeta \wedge \xi = (\omega + i\eta) \wedge (\alpha + i\beta) = (\omega \wedge \alpha - \eta \wedge \beta) + i(\eta \wedge \alpha + \omega \wedge \beta)$$

Definition 14. *The exterior derivative of a complex q-form is*

$$d\zeta = d\omega + i\,d\eta.$$

Properties of these operators extended to complex forms can be derived from the properties of their real counterparts and linearity.

The split of the tangent space into the two disjoint subspaces described in section 2.2 naturally brings additional structure to complex forms. As a consequence of eq. 4, differential forms act independently on the holomorphic and anti-holomorphic parts basis vectors and any q-form splits into a fully holomorphic part (acting on holomorphic parts of vectors) and a fully anti-holomorphic part (acting on anti-holomorphic parts of vectors). This is formalised below.

Definition 15. *Given q complex vectors V_i $(i = 1, 2, ..., q)$ either in $T_p\mathcal{M}^+$ or $T_p\mathcal{M}^-$, a complex q-form ω is said to be of bi-degree (r, s) if $\omega(V_1, V_2, ..V_q) = 0$ unless exactly r of the V_i are in $T_p\mathcal{M}^+$ and $s = q - r$ of them are in $T_p\mathcal{M}^-$. The space of (r, s) forms at p is labelled $\Omega(\mathcal{M})_p^{(r,s)}$.*

This generalises straightforwardly to form fields over $\mathcal{M}$. The bi-degree of a form corresponds to its degree in $\mathrm{d}z^\mu$ basis and its degree in $\mathrm{d}\bar{z}^\mu$. For example, $\omega = \omega_{\mu\nu\rho}\mathrm{d}z^\mu \wedge \mathrm{d}z^\nu \wedge \mathrm{d}\bar{z}^\rho$ is a 3 form of bi-degree $(2, 1)$. The orthogonality of the bases given in eq. 4 ensures that $\omega(V_i)$ with r holomorphic and s anti-holomorphic vectors V will only be non-zero if ω has r holomorphic indices and s anti-holomorphic ones.

A q form of bi-degree (r, s) in the basis defined in eq 3 is generally written

$$\omega = \frac{1}{r!s!}\omega_{\mu_1\mu_2...\mu_r\nu_1\nu_2...\nu_s}\mathrm{d}z^{\mu_1} \wedge \mathrm{d}z^{\mu_2} \wedge ... \wedge \mathrm{d}z\mu_r \wedge \mathrm{d}\bar{z}^{\nu_1} \wedge \mathrm{d}\bar{z}^{\nu_2} \wedge ... \wedge \mathrm{d}\bar{z}^{\nu_s} \qquad (5)$$

and the product of differentials $\{\mathrm{d}z^{\mu_1} \wedge \mathrm{d}z^{\mu_2} \wedge ... \wedge \mathrm{d}z^{\mu_r} \wedge \mathrm{d}\bar{z}^{\nu_1} \wedge \mathrm{d}\bar{z}^{\nu_2} \wedge ... \wedge \mathrm{d}\bar{z}^{\nu_s}\}$ forms a basis for (r, s) forms.

Not all forms are of a specific bi-degree but any q-form can be expressed as a superposition of (r, s)-forms for all (r, s) such that $q = r + s$. In practice, most of the forms we will introduce and manipulate will be of a specific (r, s) bi-degree, with the exception of some exact forms. Indeed, the exterior derivative of the general (r, s) form given in eq 5 is

$$\mathrm{d}\omega = \frac{\partial}{\partial \tilde{z}^\lambda}\omega_{\mu_1...\mu_r\nu_1...\nu_s}(z, \bar{z})\mathrm{d}\tilde{z}^\lambda \wedge \mathrm{d}z^{\mu_1} \wedge ... \wedge \mathrm{d}z\mu_r \wedge \mathrm{d}\bar{z}^{\nu_1} \wedge ... \wedge \mathrm{d}\bar{z}^{\nu_s}$$

where $\tilde{z}$ refers to both z and $\bar{z}$. Splitting the sum over the λ into both holomorphic and antiholomorphic components yields

$$\begin{aligned}\mathrm{d}\omega = &\frac{1}{r!s!}\frac{\partial}{\partial z^\lambda}\omega_{\mu_1...\mu_r\nu_1...\nu_s}(z, \bar{z})\mathrm{d}z^\lambda \wedge \mathrm{d}z^{\mu_1} \wedge ... \wedge \mathrm{d}z\mu_r \wedge \mathrm{d}\bar{z}^{\nu_1} \wedge ... \wedge \mathrm{d}\bar{z}^{\nu_s} \\ &+ \frac{1}{r!s!}\frac{\partial}{\partial \bar{z}^\lambda}\omega_{\mu_1...\mu_r\nu_1...\nu_s}(z, \bar{z})\mathrm{d}\bar{z}^\lambda \wedge \mathrm{d}z^{\mu_1} \wedge ... \wedge \mathrm{d}z\mu_r \wedge \mathrm{d}\bar{z}^{\nu_1} \wedge ... \wedge \mathrm{d}\bar{z}^{\nu_s}\end{aligned}$$

where the first and second terms are respectively $(r + 1, s)$ and $(r, s + 1)$ forms. The exterior derivative naturally splits into two operators producing forms of definite bidegrees defined below.

Definition 16. *The Dolbeault operators ∂ and $\bar{\partial}$ are defined from the exterior derivative d as $d = \partial + \bar{\partial}$. They are the linear maps*

$$\begin{aligned}\partial &: \Omega^{(r,s)}(\mathcal{M}) \longrightarrow \Omega^{(r+1,s)}(\mathcal{M}), \\ \bar{\partial} &: \Omega^{(r,s)}(\mathcal{M}) \longrightarrow \Omega^{(r,s+1)}(\mathcal{M}),\end{aligned}$$

which take $\omega \in \Omega^{(r,s)}$ *to*

$$\partial\omega = \frac{1}{r!s!}\frac{\partial}{\partial z^\lambda}\omega_{\mu_1\ldots\mu_r\nu_1\ldots\nu_s}(z,\bar{z})\,dz^\lambda \wedge dz^{\mu_1} \wedge \ldots \wedge dz\mu_r \wedge d\bar{z}^{\nu_1} \wedge \ldots \wedge d\bar{z}^{\nu_s} \in \Omega^{(r+1,s)},$$
$$\bar{\partial}\omega = \frac{1}{r!s!}\frac{\partial}{\partial \bar{z}^\lambda}\omega_{\mu_1\ldots\mu_r\nu_1\ldots\nu_s}(z,\bar{z})\,d\bar{z}^\lambda \wedge dz^{\mu_1} \wedge \ldots \wedge dz\mu_r \wedge d\bar{z}^{\nu_1} \wedge \ldots \wedge d\bar{z}^{\nu_s} \in \Omega^{(r,s+1)}.$$

Both Dolbeault operators can be shown to be nilpotent. As for real forms, exact and closed complex forms can be defined with respect to the exterior derivative. The equivalent of the de Rham cohomology class can also be constructed yielding cohomology groups of complex forms. Similarly, Dolbeault cohomologies are constructed for both ∂ and $\bar{\partial}$ operators. For example, a form obeying $\partial\omega = 0$ is said to be ∂-closed and the set of all ∂-closed (r,s) forms is labelled $Z^{r,s}_{\partial}(\mathcal{M})$. The space of exact forms and cohomology classes are constructed similarly. The equivalent for $\bar{\partial}$ can also be constructed.

Before defining Hermitian manifolds and introducing the field of Kähler geometry, we define a specific type of $(r,0)$ form:

Definition 17. *A complex form* $\omega \in \Omega^{(r,0)}(\mathcal{M})$ *is a holomorphic* $(r,0)$*-form if it obeys* $\bar{\partial}\omega = 0$*. Consequently, the components of* ω *are holomorphic functions of* z*.*

2.4 Hermitian manifolds and Kähler geometry

As explained in section 2.2, linear operators defined on a real differentiable manifold are extended to complex manifolds. This also applies to the metric.

With $X, Y, U, V \in T_p(\mathcal{M})$, a Riemannian metric g on $\mathcal{M}_{\mathbb{R}}$ is extended to $\mathcal{M}$ and acts on complexifications $Z = X + iY$, $W = U + iV \in T_p(\mathcal{M})^{\mathbb{C}}$ such that

$$g(Z,W) = g(X+iY, U+iV) = g(X,U) - g(Y,V) + i\big(g(Y,U) + g(X,V)\big) \in \mathbb{C}.$$

The metric has four types of components in the generic basis introduced in eqs. 2 and 3:

$$g_{\mu\nu} = g(\frac{\partial}{\partial z^\mu}, \frac{\partial}{\partial z^\nu}), \qquad g_{\bar{\mu}\nu} = g(\frac{\partial}{\partial \bar{z}^\mu}, \frac{\partial}{\partial z^\nu}),$$
$$g_{\mu\bar{\nu}} = g(\frac{\partial}{\partial z^\mu}, \frac{\partial}{\partial \bar{z}^\nu}), \qquad g_{\bar{\mu}\bar{\nu}} = g(\frac{\partial}{\partial \bar{z}^\mu}, \frac{\partial}{\partial \bar{z}^\nu}),$$

where we have used the convention of barred indices (i.e. $\bar{\mu}$) to distinguish between holomorphic and antiholomorphic components. Due to the defining symmetric property of g, its components aren't independent. In particular, $g_{\mu\nu} = g_{\nu\mu}$, $g_{\bar{\mu}\nu} = g_{\nu\bar{\mu}}$, $g_{\mu\bar{\nu}} = g_{\bar{\nu}\mu}$ and $g_{\bar{\mu}\bar{\nu}} = g_{\bar{\nu}\bar{\mu}}$. Additionally, we note that components of the conjugate of the metric are related to those of the metric through $\bar{g}_{\mu\bar{\nu}} = g_{\bar{\mu}\nu}$, etc.

Using the complex structure J defined on complex manifolds, a more restrictive class of metrics with additional properties can be defined by their action on the real $2m$-dim tangent space. They are then extended as described in section 2.2.

Definition 18. *A Hermitian metric g is a Riemannian metric that obeys*

$$g(J_p X, J_p Y) = g(X, Y)$$

for any $p \in \mathcal{M}$ and for any $X, Y \in T_p(\mathcal{M})$.

Hermitian metrics are therefore compatible with the complex structure J. This requirement greatly restricts the number of independent components of g once it is extended.[6] It can be easily shown (see (28), p.325) using the action of J given in definition 3 that the Hermiticity condition imposes antisymmetry on $g_{\mu\nu}$ and $g_{\bar{\mu}\bar{\nu}}$. These components are symmetric by definition and must therefore be zero. Consequently, a Hermitian metric only has $g_{\bar{\mu}\nu}$ and $g_{\mu\bar{\nu}}$ non-zero components.

Moreover, any form obeying $h_{\mu\nu} = \bar{h}_{\nu\mu}$ is referred to as Hermitian.

[6] We drop the label p on the metric from now on.

Definition 19. *A complex manifold endowed with a Hermitian metric,* $(\mathcal{M}, g)$ *is called a Hermitian manifold.*

Theorem 1. *Every complex manifold admits a Hermitian metric (28).*

The Kähler form, a particular 2-form, is then defined on Hermitian manifolds from the action of the Hermitian metric. Like the metric, it is defined on the real tangent space then extended to act on the complexified tangent space.

Definition 20. *The Kähler form of* $(\mathcal{M}, g)$ *is the* 2*-form*

$$\omega(X, Y) = g(J_p X, Y) \qquad \forall X, Y \in T_p(\mathcal{M})$$

defined smoothly at all p *in* $\mathcal{M}$*.*

When ω is extended to the complexified tangent space $T_p(\mathcal{M})^{\mathbb{C}}$, it becomes a real $(1,1)$-form. Its non-zero components are given by

$$\omega_{\mu\bar{\nu}} = ig_{\mu\bar{\nu}}$$
$$\omega_{\bar{\mu}\nu} = -ig_{\bar{\mu}\nu}.$$

The form $\omega \wedge \omega \wedge ... \wedge \omega$ where ω appears m times is a nowhere vanishing $2m$-dim real top-form (a proof can be found in (28) p. $326 - 327$). This yields the following theorem:

Theorem 2. *All complex manifolds are orientable (28).*

This guarantees that all top forms on $\mathcal{M}$ can be integrated in the usual manner. All forms differing from the Kähler form ω by an exact form (with respect to the exterior derivative) form the so-called Kähler class, of which ω is a representative.

Definition 21. *A Kähler manifold is a Hermitian manifold* $(\mathcal{M}, g)$ *whose Kähler form is closed, i.e. it obeys* $d\omega = 0$*. The associated metric* g *is a Kähler metric.*

The requirement that the Kähler form is closed allows ω to be reformulated in terms of a single real function called the Kähler potential. The statement that $\mathrm{d}\omega = 0$ is equivalent, in a coordinate patch, to

$$\partial_\alpha \omega_{\mu\bar{\nu}}(z,\bar{z})\mathrm{d}z^\alpha \wedge \mathrm{d}z^\mu \wedge \mathrm{d}\bar{z}^\nu + \bar{\partial}_{\bar{\alpha}} \omega_{\mu\bar{\nu}}(z,\bar{z})\mathrm{d}\bar{z}^\alpha \wedge \mathrm{d}z^\mu \wedge \mathrm{d}\bar{z}^\nu = 0$$

where we have used the shorthand $\partial/\partial z^\alpha = \partial_\alpha$ and $\partial/\partial \bar{z}^\alpha = \bar{\partial}_{\bar{\alpha}}$. Both terms are of different bi-degree and must therefore be 0 independently. The first term being 0 requires $\omega_{\mu\bar{\nu}} \propto \partial_\mu \beta_{\bar{\nu}}$ for $\beta_{\bar{\nu}}$ the components of a 1-form. Equivalently, the second term yields $\omega_{\mu\bar{\nu}} \propto \partial_{\bar{\nu}} \tilde{\beta}_\mu$ for $\tilde{\beta}_\mu$ the components of another 1-form. These are simultaneously satisfied by writing

$$\omega_{\mu\bar{\nu}} = i\partial_\mu \bar{\partial}_{\bar{\nu}} \mathcal{K}(z,\bar{z}) \tag{6}$$

for $\mathcal{K}(z,\bar{z}) \in \mathcal{F}(\mathcal{M})$, a real function, where the i is a conventional proportionality factor. [7] The scalar function $\mathcal{K}$ is the Kähler potential. It fully specifies the Kähler form and consequently the associated metric. This holds in a coordinate patch and the resulting expression is valid locally. To highlight this, the Kähler potential conventionally carries a patch index. For exmaple, $\omega = i\partial\bar{\partial}\mathcal{K}_c(z,\bar{z})$ in the patch $O_{(c)}$. The Kähler potential $\mathcal{K}_c(z,\bar{z})$ then transforms smoothly from patch to patch according to the transition functions. The patch index is mostly omitted in the remainder of this thesis where it is understood that the Kähler potential is defined locally.

[7] The Kähler potential is real because the Kähler form is real by definition.

2.5 Connection and curvature

Before introducing Calabi-Yau manifolds in the next section, we sketch the definitions of the metric-compatible connection on complex manifolds and the associated curvature tensor. This leads to the Ricci form and the associated first Chern class.

With a metric g, a unique connection, compatible with the complex structure can be defined.[8] We sketch the construction of the connection and refer the reader to p. $327 - 328$ of (28) for more details.

Just as with real differentiable manifolds, one can identically construct a covariant derivative acting on complex vector fields. The connection coefficients are defined from its action on basis elements, as usual, giving:

$$\nabla_\mu \frac{\partial}{\partial z^\nu} = \Gamma^\lambda_{\mu\nu}(z) \frac{\partial}{\partial z^\lambda},$$
$$\nabla_{\bar{\mu}} \frac{\partial}{\partial \bar{z}^{\bar{\nu}}} = \Gamma^{\bar{\lambda}}_{\bar{\mu}\bar{\nu}}(\bar{z}) \frac{\partial}{\partial \bar{z}^{\bar{\lambda}}},$$

where $\Gamma^\lambda_{\mu\nu}(z)$ and $\Gamma^{\bar{\lambda}}_{\bar{\mu}\bar{\nu}}(\bar{z}) = \overline{\Gamma^\lambda_{\mu\nu}(z)}$ are the connection's only non-vanishing components. Consequently, the holomorphic covariant derivative (∇_μ) acts like a partial derivative to anti-holomorphic vector fields and vice-versa.

Similarly, covariant derivatives are defined for the dual basis such that

$$\nabla_\mu \mathrm{d}z^\nu = -\Gamma^\nu_{\mu\lambda} \mathrm{d}z^\lambda,$$
$$\nabla_{\bar{\mu}} \mathrm{d}\bar{z}^{\bar{\nu}} = -\Gamma^{\bar{\nu}}_{\bar{\mu}\bar{\lambda}} \mathrm{d}\bar{z}^{\bar{\lambda}}.$$

Using the defining properties of the covariant derivative (as introduced in (29)), it can be extended to act on generic tensor fields (see (28) p. 328). Given a metric g, a connection for which $\nabla_\mu g_{\rho\bar{\sigma}} = 0$ and its conjugate hold is referred to as metric-

[8] This is analogous to the Levi-Civita connection.

compatible. A metric-compatible connection for which all components with mixed holomorphic and anti-holomorphic indices vanish is called a Hermitian connection (as was naturally constructed above). As the Levi-Civita connection, the Hermitian connection is unique. Writing down the metric-compatibility condition in terms of the connection components, their explicit form can be read off. The complex structure can then be shown to be covariantly conserved by the Hermitian connection: they are compatible structures.

Given a Hermitian connection, the Riemann curvature tensor field can be constructed. Its components take the usual form. For example, the fully holomorphic components are given by

$$R^{\kappa}_{\lambda\mu\nu} = \partial_{\mu}\Gamma^{\kappa}_{\nu\lambda} - \partial_{\nu}\Gamma^{\kappa}_{\mu\lambda} + \Gamma^{\eta}_{\nu\lambda}\Gamma^{\kappa}_{\mu\eta} - \Gamma^{\eta}_{\mu\lambda}\Gamma^{\kappa}_{\nu\eta}.$$

However, the values of the Γ components and the symmetries of the Riemann tensor restrict greatly its non-zero independent components. It can be shown ((28) p.329) that the only independent components are $R^{\kappa}_{\lambda\bar{\mu}\nu}$ and its conjugate.

Tracing over the first two indices yields the components of the Ricci tensor which, with the explicit form of the Hermitian connection components, are found to be

$$\mathfrak{R}_{\mu\bar{\nu}} = R^{\kappa}_{\kappa\mu\bar{\nu}} = -\partial_{\bar{\nu}}\partial_{\mu}\ln(\det g), \tag{7}$$

where g is the metric associated with the Hermitian connection. A metric for which the Ricci tensor vanishes is called Ricci-flat and a manifold with such a metric is a Ricci-flat manifold.

Associated to the Ricci tensor is the Ricci form, $\mathfrak{R} = i\mathfrak{R}_{\mu\bar{\nu}}\mathrm{d}z^{\mu}\wedge\mathrm{d}\bar{z}^{\bar{\nu}}$. It can be shown that it is a real form that is closed but not generally exact. The cohomology

class of the Ricci-form is called the first Chern class, $c_1(\mathcal{M}) = [\mathfrak{R}/(2\pi)]$. This is a topological invariant which characterises the non-triviality of the canonical line bundle of $\mathcal{M}$.

2.6 Calabi-Yau manifolds

With these concepts in place, we now introduce Calabi-Yau manifolds and their properties. We begin with the formal definitions of Calabi-Yau manifolds as well as the Calabi conjecture and its consequences. In the remainder of the section, we focus on the Fermat Quintic Calabi-Yau threefold, a manifold extensively used in string compactification as a warm-up to more sophisticated spaces. We detail its mathematical construction from its embedding in $\mathbb{C}P^4$. An emphasis is put on the set-up of coordinate patches and labels as this will be important in discussing the definition of the energy functional and its numerical minimisation in section 3.2.

Calabi-Yau manifolds admit different definitions which are equivalent when they are compact manifolds.[9] In line with our interest for Calabi-Yaus as Ricci-flat compactification spaces, we introduce them with the following definition.

Definition 22. *An m-dim Calabi-Yau manifold is a compact Kähler manifold which admits a Ricci-flat metric (unique in each Kahler class).*

However, this definition can be stated in many mathematically equivalent ways, in particular:

Definition 23. *An m-dim Calabi-Yau manifold is a compact Kähler manifold of vanishing first Chern class.*

[9]The story is slightly different in the case of non-compact Calabi-Yaus. This is beyond the scope of this thesis. See (10) for an accessible treatment.

An important property of m-dim Calabi-Yau manifolds sometimes also quoted as a definition, is that they always admit a nowhere-vanishing unique (up to constant rescaling) holomorphic $(m, 0)$-form. This form is denoted Ω and yields a natural volume form, $\mu = (-i)^m \Omega \wedge \bar{\Omega}$ which will be explicitly constructed for the Fermat Quintic later on.

Defs. 22 and 23 can look trivially equivalent from the definition of the first Chern class. However, we note that the Ricci-form could be non zero but exact. In that case, the Chern class would vanish but the metric would not be Ricci-flat. This was conjectured by Calabi in (30) and proved by Yau in (31; 32). Calabi famously conjectured in 1957 (30) that a vanishing first Chern class always guarantees the existence of a Ricci-flat metric. This powerful statement relating curvature to topology was only proved 20 years later by Yau in (31; 32) who showed that a Ricci-flat metric can always be constructed when the first Chern class vanishes.[10]

Further details on the Calabi conjecture and other equivalent definitions are beyond the scope of this thesis but can be found in (10).

Unfortunately, the conjecture and following proof don't provide any elements beyond the existence of the Ricci-flat metric. We note that the definitions quoted also don't point to a specific construction of Calabi-Yau manifolds. In practice, they can be easily constructed as hypersurfaces embedded in $\mathbb{C}P^{m+1}$. The construction can be generalised to products of projective spaces.[11] The specific constructions also don't directly provide an explicit Ricci-flat metric. Finding such a metric remains a topic of active research as we have described in section 1.

We refer the reader to (18; 10) for an accessible introduction to common con-

[10] Quite amazingly, Yau first attempted to disprove it by constructing (flawed) counter examples before writing down a formal proof of the conjecture.

[11] These aren't the only ways to construct Calabi-Yaus, see (10; 18) for an accessible review.

structions of Calabi-Yau manifolds and (33; 34; 35; 36) for a more formal account. The rest of this section goes into the details of the definition and set-up of the Fermat Quintic, the 3-dim Calabi-Yau manifold most often used as a starting point to show feasibility of methods.

2.7 The Quintic Calabi-Yau threefold

We first construct the complex projective space $\mathbb{C}P^N$, in which many Calabi-Yaus can be embedded.

Consider $\mathbb{C}^{N+1}$ (covered by a single coordinate patch) with coordinates labelled $\{z^a\}$ with $a = 1, 2, ..., N+1$.

Definition 24. *The N dimensional complex projective space $\mathbb{C}P^N$ is $\mathbb{C}^{N+1} - \{0\}$ with the coordinate identification $\{z^1, z^2, ...z^{N+1}\} \sim \{\lambda z^1, \lambda z^2, ...\lambda z^{N+1}\}$ $\forall \lambda \in \mathbb{C} - \{0\}$. This is the space of all undirected complex lines in $\mathbb{C}^{N+1}$ through the origin.*

The $\{z^a\}$ on $\mathbb{C}^{N+1}$ can be used to label points in $\mathbb{C}P^N$ (i.e. lines in $\mathbb{C}^{N+1}$). They are called "homogeneous coordinates" of $\mathbb{C}P^N$ and infinitely many $\{z^a\}$ (all equivalent under $\sim$ defined above) correspond to a single point in $\mathbb{C}P^N$. Alternatively, an "inhomogeneous coordinate system" (where a point is specified by a unique set of coordinates) can be defined on $\mathbb{C}P^N$ using at least $N+1$ coordinate patches.

The following is illustrated in fig. 1 with $N = 4$. A non-zero coordinate in $\{z^a\}$, labelled z^c, defines the patch $O_{(c)}$ on $\mathbb{C}P^N$. It contains all points with $z^c \neq 0$. In this patch, a point is labelled by the coordinates

$$u^a_{(c)} = \frac{z^a}{z^c} \text{ with } a = 1, 2...N+1.$$

This includes $u^c_{(c)} = 1$. For cleanliness, $u^c_{(c)}$ is usually dropped and the set of coor-

dinates, labelled by Greek indices, now contains N elements:

$$u^\alpha = \frac{z^\alpha}{z^c} \text{ with } \alpha = 1, 2...N \text{ skipping c.}$$

The patch label c is omitted from now on for ease of notation. This requires at least $N + 1$ patches, overlapping on points with multiple non-zero coordinates z^a.

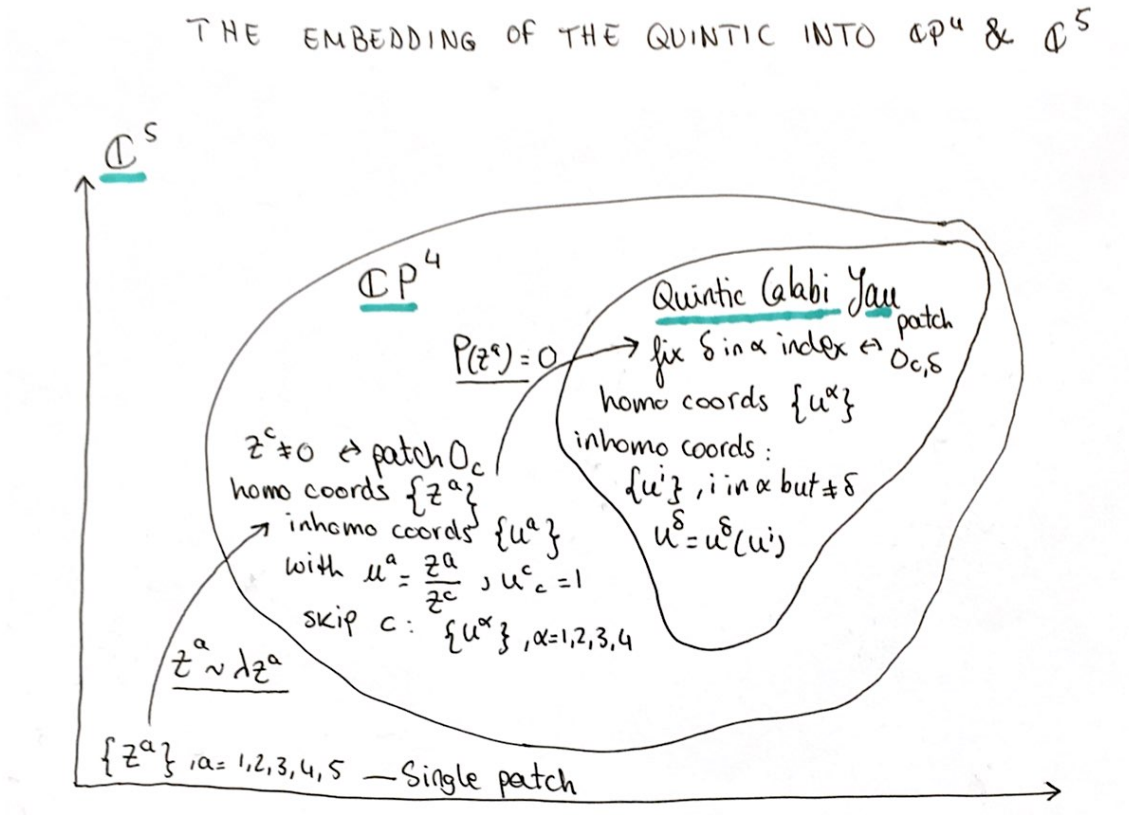

Figure 1: The embedding of $\mathbb{C}P^4$ and the Calabi-Yau Quintic in $\mathbb{C}^5$ is schematically represented. The relationships between the coordinates systems on those spaces are illustrated.

A natural family of metrics on $\mathbb{C}P^N$ are the generalised Fubini-Study metrics, which will be the starting point when attempting to find the Calabi-Yau metric later on. Fubini-Study metrics are defined from the form of their Kähler potential

as follows.

Definition 25. *Given a positive definite Hermitian form $G_{a\bar{b}}$ on $\mathbb{C}^{N+1}$, the Fubini-Study metric on $\mathbb{C}P^N$ is the metric whose Kähler potential is of the form*

$$\mathcal{K}^{FS} = \ln\left(G_{a\bar{b}} u^a \bar{u}^{\bar{b}}\right) \tag{8}$$

locally.

Different Hermitian forms G pick out particular representatives in the Kähler class which is determined by the embedding in $\mathbb{C}P^N$. The class of the Kähler form ω therefore does not depend on G.

With the coordinates defined above, the Fermat Quintic, X is defined in $\mathbb{C}^5 - \{0\}$ as the locus of points which obey

$$P(z) = \Sigma_{a=1}^{5} (z^a)^5 = (z^1)^5 + (z^2)^5 + (z^3)^5 + (z^4)^5 + (z^5)^5 = 0. \tag{9}$$

This is illustrated in fig. 1. The defining eq. 9 can be rewritten in coordinates on $\mathbb{C}P^4$, giving

$$P(u) = 1 + \Sigma_{\alpha=1}^{4} (u^\alpha)^5 = 0 \tag{10}$$

Coordinate patches on X also can be constructed. [12] A coordinate u^δ in $\{u^\alpha\}$ ($\delta \neq c$, of course) is fixed by expressing it as a function of the remaining coordinates using eq. 10. This patch, labelled $O_{(c,\delta)}$, is characterised by the remaining three coordinates $\{u^i\}$ where $i = 1, 2, 3$ (skipping δ). This forms inhomogeneous coordinates on the Quintic. These different coordinate systems and embeddings are illustrated in fig. 1. As expected, the Fubini-Study Kähler potential keeps its form when pulled back to

[12] However, in practice using them avoided where possible.

X but only admits three independent arguments when evaluated on X.

In practice, it is much easier to work in $\mathbb{C}P^4$ with the additional constraint $P(u) = 0$ to obtain points on X than to build coordinate patches on the Calabi-Yau threefolds. This requires caution when using derivatives. To take derivatives on X, the polynomial P is kept fixed while it is not when taking derivatives on $\mathbb{C}P^4$. These derivatives are respectively labelled $\hat{\partial}_i$ and ∂_i (or ∂_δ when taking a derivative with respect to the additional $\mathbb{C}P^4$ coordinate). A relationship between them is derived using the chain rule in section 3.2.4.

3 Computing Calabi-Yau metrics

Calabi-Yau manifolds admit a Ricci-flat metric (sometimes called a Calabi-Yau metric), which solves Einstein's vacuum equations. But neither the Calabi conjecture nor the explicit constructions of the manifold specify how to find the Ricci-flat metric. There has been no successful attempt at analytically writing down Ricci-flat metrics. [13] Finding Calabi-Yau metrics is now mainly approached as a numerical problem (22) and progress was made using various approaches (3; 23; 24; 1).

In this section, we present two methods to numerically compute Calabi-Yau metrics. We describe in broad terms Donaldson algorithm and highlight recent work where machine learning was used to mitigate some of its drawbacks. We then introduce in detail an alternative method, based on the construction and minimisation of an energy functional over the space of Kähler metrics. This was originally developed by Headrick and Nassar in (1). Building up from their work, we then offer a hybrid method using supervised machine learning implemented through a neural network.

[13] Except on highly symmetric Calabi-Yaus presenting little interest as a compactification space, see (36; 22) for more details

Using quantities computed with Headrick and Nassar's `Mathematica` package available at (37), one seeks to replace their minimisation process to predict the Kähler potential with one requiring fewer computational resources. The latter is original work performed by the author under the supervision of Prof Yang-Hui He and Dr Anthony Ashmore.

3.1 Donaldson algorithm

Donaldson algorithm, like the energy functional method of (1) that we introduce in section 3.2, encodes the metric information numerically using "algebraic metrics".[14] The construction, originally attributed to (38) is reproduced in detail in sections 3.2.1 and 3.2.2 when introducing the energy functional method. The main idea is to express the Kähler potential as an expansion in a complete basis, commonly the eigenspaces of the Laplacian on $\mathbb{C}P^4$. These eigenspaces are built from homogeneous polynomials of degree k and form a complete basis on $\mathbb{C}P^4$. When pulled back to the Calabi-Yau X, the polynomials aren't linearly independent anymore. A complete basis on X is obtained by removing all redundant polynomials.

The algebraic metrics are then constructed from the Kähler potential expanded in the basis. The coefficients of superposition encode the metric information. The accuracy of the results (and the number of coefficients) then increases with k. Consequently, the larger k, the closer to "Ricci-flatness" is the resulting Calabi-Yau metric and the longer the computations to find the metric become.

Using this set up and theoretical work of Donaldson (2), a numerical procedure to find an approximate Calabi-Yau metric was presented in (3; 25). The algorithm

[14] A similar encoding of the information was presented in (24) using a symplectic structure and coordinates.

relies on a non-linear map, the "T map", that acts on the coefficients. This map admits a unique fixed point (such that $T(h) = h$) which corresponds to the "balanced metric". Donaldson showed in (2), that iteratively applying the map to the coefficients, the metric converges to the balanced metric. This balanced metric is itself an approximation to the Ricci-flat metric. Extended details of the procedure are beyond the scope of this thesis but can be found in the original work (2; 3; 25) and in introductory sections of (4).

The power of this method is the small number of iterations (10 or so) necessary to reach convergence (to sufficient accuracy for most applications). However, the algorithm relies on the generation of a significant number of points (of the order of 10^9 for $k = 8$), which is a time expensive process. The required number of points also increases with k, further limiting the best reachable accuracy. Additionally, while the increasing accuracy was originally predicted to increase exponentially with k, the metrics were found to only approach Ricci-flatness as a power of k (1). Overall, this method produces metrics that are Ricci-flat within 1% error which is sufficient for most applications. The time required is of the order of hours and the storage requirements are fewer, thus providing a net improvement compared to previous attempts (23). In particular, for the Quintic, computing the metric for $k = 12$ was found to take 50 hours (4).[15] While a 50 hours computation is manageable once or twice, this becomes a significant limitation when one needs to scan over a few Calabi-Yaus.

The limitations of Donaldson algorithm can be mitigated in different ways. One approach is to change the modality of some of its steps. In particular (4) showed that it is possible to use supervised learning, a form of machine learning, to significantly

[15] A back of the envelope calculation in (4) extrapolates this to suggest that $k = 20$ would then require about 35 years to yield results...

decrease the computation time to reach a metric. They managed to reduce the required time by a factor of 50 and produce metrics of a similar accuracy in an hour. The method can be summarised as follows.

1. A set of points on the Calabi-Yau (larger than the number necessary for convergence) is computed (they use $500,000$).

2. Coefficients in the polynomial expansions are then calculated for $k = 4, 5, 6, 7$ using the smallest subset of the points guaranteeing convergence (that turns out to be $99,000$).

3. Using the resulting coefficients, the metrics are calculated on the remaining points.

4. Through curve-fitting techniques, the metric can then be extrapolated to higher k (they do this up to $k = 15$) on a subset of the points (they use $10,000$ points).[16] These metrics are of lower accuracy than when calculated through Donaldson algorithm.

5. Using supervised learning, a machine learning algorithm is then trained on the values of the metric calculated with Donaldson's at $k = 4, 5, 6, 7$ and the values of the metrics extrapolated on $10,000$ points.

6. Upon successful training (which (4) demonstrates) the algorithm can then predict the metric at $k = 15$ can then be fully predicted for the $500,000$ points computed initially.

They found that their final metrics computed using the method above were similar in accuracy to Donaldson's $k = 12$ metrics. The former takes an hour for the full metric

[16] This is known as extrapolating to the continuum.

components while the latter needs about 50 hours. This highlights the significance of the time gain with minimal compromises on the accuracy of the results. This approach was investigated in detail for the Fermat Quintic and deemed a success. It can be directly applied to other Calabi-Yau threefold, for example the torus-fibred Schoen Calabi–Yau threefolds used in (39) to obtain a heterotic standard model. This work is currently underway by the collaboration behind (4).

Alternatively, a radically different method was developed by Headrick and Nassar in (1). They encode the numerical information using the same algebraic metrics but instead of finding a balanced metric using the "T map", they construct an energy functional that admits a unique minimum on the Ricci-flat metric. This turns the numerical Calabi-Yau metric problem into a minimisation procedure, which is well studied and easy to implement and solve. The method requires significantly less points than Donaldson's method. Moreover, the number of point required doesn't increase significantly with k. In practice, Headrick and Nassar found that 3000 is sufficient to obtain metrics of similar accuracy (about 1% deviations from Ricci-flatness). The accuracy of the metrics was also shown to improve exponentially with k.

Overall, the method is easy to implement, requires less storage and computational power and yields results of higher accuracy. It thus presents significant improvements to Donaldson algorithm.

A drawback of this method is that it relies on the exploitation of the symmetries of the manifold. This is used to reduce the number of polynomial basis elements. This is particularly well suited to produce metrics on the highly symmetric Quintic. Results of high accuracy are then produced in short times.[17] However, this trick

[17]Headrick and Nassar claim in (23) that a few minutes of computation yield results of sufficient accuracy for most applications. We present an in-depth analysis of the computation times in

can't be generalised to more complicated Calabi-Yau spaces. This would require ignoring the symmetries of the space. This consequently increases the dimension of the basis elements and the space of Kähler metrics that is minimised over, which in turn translates to longer computation times. This currently significantly limits the use of the method to obtain the metric on other Calabi-Yaus, in particular the ones relevant to string theory.

This limitation can be mitigated by speeding up the overall method to a point where the additional time required when ignoring symmetries becomes manageable. This is the goal we work towards in the remainder of this thesis. Building up on the work using machine learning on data collected with Donaldson algorithm presented in (4), we propose a similar approach to Headrick and Nassar's method. This is explored in section 4 where we replace their ultimate minimisation process by a neural network to shorten the required computation time.

Before we introduce the specifics of our method and present our results, we give details of the energy functional method and show why it works. We use Headrick and Nassar's strategy document (40) to gain an understanding of (1) and provide here an accessible description of the method.

3.2 The energy functional minimisation method

A word about conventions: we chose to stick to the standard labelling for the complex structure (J), Kähler form (ω) and Calabi-Yau holomorphic $(m, 0)$ form (Ω). However, this is different from the conventions used by Headrick and Nassar in (40) and (1) where the Kähler form is labelled J. We hope this isn't a source of confusion and have tried to make the following section as self-contained as possible to reduce

section section 3.2.6.

any back and forth with (40) and (1).

We start by defining the polynomial basis used to numerically express the Kähler potential. Most of the construction is done in homogeneous coordinates on $\mathbb{C}^5$ and $\mathbb{C}P^4$ to avoid using coordinate patches on X. Once this is set up, the energy functional is defined and massaged into a form appropriate for the minimisation. Finally, the algorithm is summarised.

3.2.1 Polynomial basis

As briefly mentioned in section 3.1, the Kähler metrics on X are constructed by expanding their corresponding Kähler potential in a complete basis. A conventional choice of basis is the eigenspaces of the Laplacian operator. While these are tedious to construct on X, they take the simple form of polynomials on $\mathbb{C}P^N$. This is the basis that is used to represent the potential. We have chosen to introduce the method using $N = 4$ exclusively, which corresponds to the Quintic Calabi-Yau. The same construction could be done for a general N (as is presented originally in (1)) and applied to Calabi-Yau hypersurfaces of different dimensions.

Let $\rho^I(z)$ be the set of homogeneous polynomials of degree k in the homogeneous coordinates z^a where the index I iterates over the elements. For example, for $k = 1, 2$ the sets are given by

$$k = 1 : \{z^1, z^2, z^3, z^4, z^5\},$$

$$k = 2 : \{z^1z^1, z^1z^2, z^1z^3, z^1z^4, z^1z^5, z^2z^2, z^2z^3, z^2z^4, z^2z^5, z^3z^3, z^3z^4, z^3z^5, z^4z^4, z^4z^5, z^5z^5\}.$$

The sets have dimensions given by $N_k = (N + k)!/(N!k!)$ where N is the dimension of the projective space we work in (for our purposes, $N = 4$). The eigenspaces of the Laplacian, up to k, in the patch $O_{(c)}$, are then spanned by the functions

$$\frac{\rho^I(z)\bar{\rho}^{\bar{J}}(\bar{z})}{(G_{a\bar{b}}z^a\bar{z}^{\bar{b}})^k} \tag{11}$$

where $G_{a\bar{b}}$ is the Hermitian form specifying the Fubini-Study metric on $\mathbb{C}P^4$. We highlight that ρ^I and $\bar{\rho}^{\bar{J}}$ are taken at a given k.

While eq. 11 provides a basis on $\mathbb{C}P^4$, the basis elements aren't linearly independent anymore when pulled back to X. To obtain a basis on X the polynomials must be quotiented out by the defining Quintic equation ($P(z) = 0$, eq. 9). Mathematically, this corresponds to removing from ρ^I any degree $(k-5)$ polynomial proportional to P (defined in eq. 9). All homogeneous polynomials of degree k linearly independent on X are denoted p^A. The dimensionality of p^A is then $N'_k = N_k - N_{(k-5)}$ for $k > 5$ and $N'_k = N_k$ for $k \leq 5$. Different indices (I and A) are used to highlight the different dimensions.

3.2.2 Kähler potentials

We now describe how Kähler potentials are expressed in terms of the eigenspaces defined in eq. 11.

The Calabi conjecture implies that there exists a unique Ricci-flat metric in every Kähler class. In the case of the Quintic, there is only one cohomology class as $h^{(1,1)} = 1$ (all closed $(1,2)$-forms differ by an exact form). As a consequence, this Calabi-Yau admits a single Ricci-flat metric. This isn't necessarily the case for more sophisticated Calabi-Yaus for which every Kähler class would contain a unique Ricci-flat representative. In any case, this method finds the Ricci-flat representative in the class containing the Fubini-Study metric which serves as starting point when minimising the functional over the space of possible Kähler potentials.

Any representative with potential $\mathbb{K}$ in the same Kähler class as $\mathcal{K}^{FS}$ differs by a globally defined function. In practice, the exponential of this function is expanded in the basis as follows:

$$e^{k(\mathcal{K}-\mathcal{K}^{FS})} = h_{A\bar{B}} \frac{p^A(z)\bar{p}^{\bar{B}}(\bar{z})}{(G_{a\bar{b}} z^a \bar{z}^{\bar{b}})^k},$$

where $h_{A\bar{B}}$ is a positive definite Hermitian $N'_k \times N'_k$ matrix. Taking the logarithm of both sides and substituting the form of $\mathcal{K}^{FS}$, given in eq. 8 yields

$$\mathcal{K} = \frac{1}{k} \ln \left(h_{A\bar{B}} p^A(z) \bar{p}^{\bar{B}}(\bar{z}) \right) - \ln \left(|z^c|^2 \right)$$

which, although we omit the c label on the potential for readability, is only valid in the patch $O_{(c)}$. Expressing the polynomials in terms of the coordinates u^a on $\mathbb{C}P^4$ then yields the following expression

$$\begin{aligned} \mathcal{K} &= \frac{1}{k} \ln \left(h_{A\bar{B}} p^A(u) \bar{p}^{\bar{B}}(\bar{u}) \right) \\ &= \frac{1}{k} \ln \psi \end{aligned} \tag{12}$$

(still valid in the patch $O_{(c)}$), where $\psi = h_{A\bar{B}} p^A(u) \bar{p}^{\bar{B}}(\bar{u})$.

We point out that $p^A(u)$ takes as argument the u^a, which includes $u^c = 1$ in the given patch. For a given k, the product $p^A(u)\bar{p}^{\bar{B}}(\bar{u})$ will contain elements of degree k in both u^α and the corresponding complex conjugate but also elements of lower degree. For example, taking $k = 1$ and specifying $c = 5$, the Kähler potential is expanded in the members of $p^A \times \bar{p}^{\bar{B}} = \{u^1, u^2, u^3, u^4, 1\} \times \{\bar{u}^1, \bar{u}^2, \bar{u}^3, \bar{u}^4, 1\}$. Multiplying the elements out, the set product contains $1, u^1, u^2, ..., \bar{u}^1, ..., u^1\bar{u}^2, ...$ and confirms that all degrees up to k are present in the expansion although the polynomials in p^A were taken at k only. This straightforwardly applies to $k > 1$.

Different matrices $h_{A\bar{B}}$ pick out specific representatives in the Kähler class. When looking for the Ricci-flat metric, these are the coefficients that are minimised for. In practice, $\mathcal{K}$ is relevant up to constant scaling as the metric is determined from its derivatives. Therefore $h_{A\bar{B}}$ is computed up to overall scaling to avoid redundancy. The dimension[18] of $h_{A\bar{B}}$ corresponds to the dimension of a subset of the Kähler class that can be practically scanned for the most Ricci-flat metric when optimising the coefficients $h_{A\bar{B}}$. As k increases, the subset tends to the full space of Kähler metrics. It was shown in (1) that $k = 3$ yielded accurate enough results for most applications. We therefore deduce that the reduction of the Kähler class to this subset doesn't have a significant effect.

The advantage of these algebraic metrics is the simplicity of the construction in terms of polynomials. It only requires coordinate patches in $\mathbb{C}P^4$ which are all identical and simple to manipulate. However, the size of the matrix $h_{A\bar{B}}$ grows extremely quickly with k, about $\sim k^{2(N-1)} = k^6$ in the case of the Quintic. The Quintic however admits an internal $\mathbb{Z}^5 \times \mathbb{Z}^5$ symmetry and consequently any metric and thus Kähler potential must be invariant under these transformations. This allows for reducing the basis of eigenspaces given in eq. 11 by quotienting it by the discrete symmetry group of the Quintic. This group given by all transformation leaving P invariant contains the following operations expressed in the $\mathbb{C}^5$ coordinates in X: permutations of any z^a, overall complex conjugation ($z^a \to \bar{z}^a$), multiplication of any z^a by the fifth root of unity (modulo overall multiplication by any root of unity).

Instead of reducing the size of the polynomials in p^A, Headrick and Nassar limit the number of terms appearing in ψ and they simultaneously reduce the overall basis

[18]By dimension we mean the number of degrees of freedoms, not the size.

to X. This approach yields an alternative formulation where they construct directly a complete and linearly independent basis on X that is also invariant under the symmetry group of X. In this formulation, basis elements (corresponding to eq. 11) become vectors given by

$$\mathcal{P}^l = c^l_{I\bar{J}} \bar{\rho}^{\bar{J}} \rho^I. \tag{13}$$

The matrix $h_{A\bar{B}}$ is then replaced by h_l such that $\psi = h_l \mathcal{P}^l$. The multi-indices object $c^l_{I\bar{J}}$, found with `Mathematica` (41), restricts the basis to X. The metric information is now encoded in h_l where l iterates over the basis elements as opposed to a pair of holomorphic and anti-holomorphic indices as in eq. 11. In that case, the values of h_l are the parameters that are varied when minimising the functional defined in the next section. This corresponds to varying the Kähler metric within its class. The metric closest to Ricci-flat is thus found by scanning over the metrics this way.

To compute the results in (1), the second formulation was used and this is what is implemented in Headrick and Nassar's Mathematica (41) package available online at (37).[19] In the description of the energy functional method in (1) and (40), Headrick and Nassar mainly use the first formulation which doesn't account for the symmetries of the Calabi-Yau. To facilitate any reference to(40), we use the first formulation here when presenting the functional in the following section. Once the method is set up, we reproduce an argument, originally given in (40) which shows that both formulations which are essentially equivalent.

[19]We note that this formulation can also be used without exploiting the symmetries of the manifold at hand. This affects the form of $c^l_{I\bar{J}}$.

3.2.3 Energy functional

With the representation of the Kähler potential in terms of the eigenspaces of the Laplacian given in the previous section, the energy functional to be minimised can now be defined.

The geometrical structure of Calabi-Yau manifolds yields two natural volume forms. The first one is constructed through wedge products of the Kähler form – $\omega \wedge \omega \wedge \omega$ – and the second one using the natural holomorphic form the Calabi-Yau $\mu = (-i)^3 \Omega \wedge \bar{\Omega}$. Their ratio is formally given by

$$v_\omega = \frac{\omega \wedge \omega \wedge \omega}{m! \mu}, \tag{14}$$

where we use the subscript ω to highlight the dependence on the Kähler representative ω. It should be understood as the ratio of their single component in a given orientation. The ratio can be manipulated into the form

$$v_\omega = \frac{\det g_{i\bar{j}}}{|\Omega_{123}|^2} \tag{15}$$

where $g_{i\bar{j}}$ is the metric associated to ω on X in inhomogeneous coordinates.

As defined in section 2.5, the Ricci tensor associated to the metric $g_{i\bar{j}}$ has components given by

$$\mathcal{R}_{i\bar{j}} = -\partial_i \bar{\partial}_{\bar{j}} \ln (\det g_{i\bar{j}}). \tag{16}$$

Recalling that Ω is a holomorphic form, $\bar{\partial}_{\bar{j}} \Omega_{123} = 0$. The quantity $\partial_i \bar{\partial}_{\bar{j}} \ln v_\omega$ differs from $\mathcal{R}_{i\bar{j}}$ by $\partial_i \bar{\partial}_{\bar{j}} \ln (\Omega_{123} \bar{\Omega}_{123})$ which vanishes by the holomorphic property of Ω.

The Ricci tensor components originally defined in eq. 16 are therefore equivalent to

$$\mathcal{R}_{i\bar{j}} = -\partial_i \bar{\partial}_{\bar{j}} \ln v_\omega .$$

The unique representative ω in the Fubini-Study Kähler class for which the Ricci tensor vanishes then obeys $v_\omega = \text{constant}$ or $\omega \wedge \omega \wedge \omega \propto \Omega \wedge \bar{\Omega}$. The latter form is an instance of the Monge-Ampère equation. Without loss of generality, the scaling factor can be set such that $v_\omega = 1$ on the Ricci-flat representative. Since the components of the Ricci-tensor are determined by derivatives of $\ln v_\omega$, scalar multiplication of v_ω leaves the components unchanged

Finding a Ricci-flat metric hence reduces to picking a representative ω such that $v_\omega = 1$. Having expressed the Kähler potential in the p^A basis (in the first formulation), the task consists in varying $h_{A\bar{B}}$ to find the right ω (or equivalently, h_l in the second formulation). This is done by minimising an appropriate energy functional over the space of $h_{A\bar{B}}$ matrices. Headrick and Nassar present, in particular, two such functionals with the key properties that they admit a unique minimum on the Ricci-flat metric, have no other critical points and are non-negative. Both vanish on the Ricci-flat representative only.

The two functionals are

$$\begin{aligned} H_1[\omega] &= \int_X \mu \Big(v_\omega - \frac{V_\omega}{V_\mu} \Big)^2 , \\ H_2[\omega] &= -\frac{1}{2} \int_X \mu R_\omega \end{aligned} \tag{17}$$

where $R_\omega = g^{i\bar{j}} \partial_i \ln (v_\omega) \bar{\partial}_{\bar{j}} \ln (v_\omega)$ is the Ricci scalar of the Kähler metric $g_{i\bar{j}}$. The quantities V_ω and V_μ are the volumes of X with respect to the two volume forms

defined previously. Namely,

$$V_\omega = \int_X \mu v_\omega \propto \int_X \omega \wedge \omega \wedge \omega \tag{18}$$

and

$$V_\mu = \int_X \mu. \tag{19}$$

It can be shown that V_ω only depends on the Kähler class, not the representative ω. It is therefore be computed once and remains fixed during the minimisation.

The first functional corresponds to the "variance" of v_ω over the manifold. Quotation marks are used as the variance is weighted by the measure μ and is therefore not equivalent to the statistical variance of the integrand viewed as a statistical sample. The integration therefore can't be approximated by the statistical variance of the integrand and needs to be performed. The meaning of H_2 is less straightforwardly explained. Its form appears very similar to the Einstein-Hilbert action, which by construction, admits a minimum on the solutions of Einstein's equations. It differs from H_2 by the volume form integrated over. The functional defined here is integrated over μ whereas the Einstein-Hilbert action would be integrated over the volume form constructed from the Kähler form. This difference is what makes H_2 suitable since the Einstein-Hilbert action can be shown to vanish for any Kähler metric (1). A more detailed discussion of the relevance and motivations behind H_2 can be found in (1).

Other functionals could be constructed to admit a minimum on the Ricci-flat metric and the two presented here are simply a starting point. From their form, $H_1[\omega]$ is significantly easier to compute as it doesn't require the gradient information necessary to calculate $H_2[\omega]$. From now on we will work with H_1, henceforth labelled

H, to obtain the Ricci flat metric.

3.2.4 The minimisation

The elements of the previous sections are now brought together to show exactly how the minimisation process yields the Ricci-flat metric. The key is to reformulate the integrand to render the dependence on $h_{A\bar{B}}$ explicit. We wish to exclusively work in the $\mathbb{C}P^4$ embedding of the Calabi-Yau. All quantities are therefore expressed in the u^α homogeneous coordinates on X. We say a few words about the integration and generation of the points but mainly refer the reader to p.7 of (40) for more rigorous details which are beyond the understanding required to follow this thesis.

The integrand

The following 4 component object is defined

$$Q_\alpha = \frac{\hat{\partial} P}{\partial u^\alpha}\Big|_{u^\beta;\beta\neq\alpha} \tag{20}$$

in the patch $O_{(c)}$ on $\mathbb{C}P^4$. As denoted by $\hat{\partial}$, the derivative is taken on $\mathbb{C}P^4$ as defined in section 2.7. Pulling back Q_δ to the patch $O_{(c,\delta)}$ in X, the $(3,0)$ non-vanishing Calabi-Yau holomorphic form Ω (unique up to a constant scaling) is defined as

$$\Omega = Q_\delta^{-1} \prod_i \mathrm{d}u^i$$

in inhomogeneous coordinates on X. We abbreviate $\mathrm{d}u^1 \wedge \mathrm{d}u^2 \wedge \mathrm{d}u^3$ by $\prod_i \mathrm{d}u^i$, a convention used throughout. From this, it can be inferred that $\Omega_{123} = Q_\delta^{-1}$ (ignoring

combinatorial factors). The volume form is then

$$\mu = (-i)^3 |Q_\delta|^{-2} \prod_i \mathrm{d}u^i \wedge \prod_{\bar{j}} \mathrm{d}\bar{u}^{\bar{j}} \tag{21}$$

on X.

Another ingredient required to express v_ω in terms of homogeneous coordinates on $\mathbb{C}P^4$ is the relationship between derivatives on X, ∂_i, and on $\mathbb{C}P^4$, $\hat{\partial}_i$. As mentioned in section 2.7, derivatives on X are at u^j ($j \neq i$) and P fixed, implying that u^δ, the coordinate on $\mathbb{C}P^4$ (that becomes a scalar function on X) is implicitly varied. On the other hand, derivatives in $\mathbb{C}P^4$ are at both u^j and u^δ fixed (since u^δ is a simple coordinate, on the same footing as u^j). This gives, using the chain rule on X,

$$\frac{\partial}{\partial u^i}\Big|_{u^j} = \frac{\partial}{\partial u^i}\Big|_{u^j,u^\delta} + \frac{\partial u^\delta}{\partial u^i}\Big|_{u^j} \frac{\partial}{\partial u^\delta}\Big|_{u^i}, \tag{22}$$

where we identify $\hat{\partial}_i \equiv \frac{\partial}{\partial u^i}\Big|_{u^j,u^\delta}$ and $\hat{\partial}_\delta \equiv \frac{\partial}{\partial u^\delta}\Big|_{u^i}$.

An expression for $\frac{\partial u^\delta}{\partial u^i}\Big|_{u^j}$ is found by applying eq. 22 to P, remembering that it vanishes identically on X. This yields, in the patch $O_{(c,\delta)}$ on X (a trick found in (3)),

$$\frac{\partial P}{\partial u^i}\Big|_{u^j} = \frac{\partial P}{\partial u^i}\Big|_{u^j,u^\delta} + \frac{\partial P}{\partial u^\delta}\Big|_{u^i} \frac{\partial u^\delta}{\partial u^i}\Big|_{u^j} = 0.$$

Using the definition of Q_α and rearranging gives

$$\frac{\partial u^\delta}{\partial u^i}\Big|_{u^j} = -\frac{Q_i}{Q_\delta}.$$

Plugging this into eq. 22 gives us the final relationship between the derivatives:

$$\partial_i = \hat{\partial}_i - \frac{Q_i}{Q_\delta}\hat{\partial}_\delta. \tag{23}$$

This relationship between derivatives translates into a relationship between the metrics on $\mathbb{C}P^4$ and X, $g_{i\bar{j}} = \partial_i \bar{\partial}_{\bar{j}} \mathcal{K}$ and $\hat{g}_{\alpha\bar{\beta}} = \hat{\partial}_\alpha \hat{\bar{\partial}}_{\bar{\beta}} \mathcal{K}$ respectively. In particular, using matrices and determinant identities, it is found that

$$\det g_{i\bar{j}} = \frac{|Q|^2}{|Q_\delta|^2} \det \hat{g}_{\alpha\bar{\beta}} \tag{24}$$

where $|Q|^2 = \hat{g}^{\alpha\bar{\beta}} Q_\alpha \bar{Q}_{\bar{\beta}}$. The ratio v_ω as written in eq. 15 can then be reformulated as

$$v_\omega = |Q|^2 \det \hat{g}^{\alpha\bar{\beta}}, \tag{25}$$

now expressed fully through the embedding of X in $\mathbb{C}P^4$.

Using matrix identities, we now explicitly write $\det \hat{g}^{\alpha\bar{\beta}}$ in terms of $h_{A\bar{B}}$ from the definition of the metric in terms of the Kähler potential. We defie the following quantities:

$$\begin{aligned}
q^A_\alpha &= \hat{\partial}_\alpha p^A(u), \\
\bar{q}^{\bar{B}}_{\bar{\beta}} &= \hat{\bar{\partial}}_{\bar{\beta}} \bar{p}^{\bar{B}}(\bar{u}), \\
\psi_\alpha &= h_{A\bar{B}} q^A_\alpha(u) \bar{p}^{\bar{B}} = \hat{\partial}_\alpha \psi, \\
\psi_{\bar{\beta}} &= h_{A\bar{B}} p^A(u) \bar{q}^{\bar{B}}_{\bar{\beta}} = \hat{\bar{\partial}}_{\bar{\beta}} \psi, \\
\psi_{\alpha\bar{\beta}} &= h_{A\bar{B}} q^A_\alpha \bar{q}^{\bar{B}}_{\bar{\beta}} = \hat{\partial}_\alpha \hat{\bar{\partial}}_{\bar{\beta}} \psi.
\end{aligned} \tag{26}$$

Applying the product rule to the Kähler potential defined in eq. 12, the metric then

takes the form, with the variables defined above,

$$k\hat{g}_{\alpha\bar{\beta}} = \Big(\frac{\psi_{\alpha\bar{\beta}}}{\psi} - \frac{\psi_{\alpha}\psi_{\bar{\beta}}}{\psi^2}\Big). \tag{27}$$

For convenience we repackage the quantities defined in eq. 26 into 5×5 matrices and 5-component vectors as follows

$$\begin{aligned} \Psi_{a\bar{b}} \quad & \begin{pmatrix} \psi_{\alpha\bar{\beta}} & \psi_{\alpha}, \\ \psi^{T}_{\bar{\beta}} & \psi \end{pmatrix} \\ Q_a &= (Q_\alpha, 0), \end{aligned} \tag{28}$$

where $a = \alpha, c$ in the patch $O_{(c)}$, i.e $\Psi_{\alpha c} = \psi_\alpha$ and $\Psi_{c\bar{c}} = \psi$. Using Schur's determinant identity together with the first few formulae in (42) and taking into account the symmetry properties of the quantities manipulated, it can be verified that

$$v_\omega = k^{-5}\psi^{-4} \det \Psi \bar{Q}_{\bar{b}} \Psi^{\bar{b}a} Q_a, \tag{29}$$

where the dependence on ψ and hence $h_{A\bar{B}}$ is now explicit. This expression for v_ω doesn't depend on the metrics and all quantities are expressed in $\mathbb{C}P^4$ coordinate patches. This is the form of v_ω that is calculated at random points in X, integrated, and then optimised for the most Ricci-flat ψ.

Going back to the second formulation, v_ω and its building blocks are now expressed in terms of the vector basis $\mathcal{P}^l$ defined in eq. 13. The corresponding optimisation coefficients become h_l. We have already noted that $\psi = h_l \mathcal{P}^l$. The quantities

defined in eqs. 26 and 28 become

$$\Psi_{a\bar{b}} = h_l \mathcal{Q}^l_{a\bar{b}}, \tag{30}$$

with

$$\begin{aligned} q^I_c &= \rho^I, \\ q^I_\alpha &= \hat{\partial}_\alpha \rho^I, \\ \mathcal{Q}^l_{a\bar{b}} &= c^l_{I\bar{J}} \bar{q}^{\bar{J}}_{\bar{b}} q^I_\alpha. \end{aligned} \tag{31}$$

In their `Mathematica` implementation (available in the `Fermat.m` package and `optimal.nb` notebook at (37)), these are the quantities that Headrick and Nassar are working with. In this language, the coefficients h_l are minimised for.

Numerical integration

The integral in eq. 17 is evaluated using a Monte-Carlo method. It is approximated by a sum of the integrand, weighted by the volume form μ, evaluated at random points generated with respect to the measure μ. The overall method to find Calabi-Yau metrics presented here doesn't rely on the particular method to generate the points for the integration. In this section, we therefore give a flavour of the particular algorithm used by Headrick and Nassar to generate the points and refer the reader to (40) for a more rigorous account of this part of their work.

The volume form μ in the form given by eq. 21 is expressed in homogeneous coordinates u^α on X embedded in $\mathbb{C}P^4$ by using the integral definition of the delta function (over $\mathrm{d}u^\delta$). We find that

$$\mu = (-i)^3 \delta(P)^2 \prod_\alpha \mathrm{d}u^\alpha \wedge \prod_{\bar{\beta}} \mathrm{d}\bar{u}^{\bar{\beta}}. \tag{32}$$

This corresponds to the same volume form (when integrated over), written in $\mathbb{C}P^4$ and restricted to X.

To generate the points, Headrick and Nassar spread the delta function by $\pm\epsilon$ (which can be set to 0.01 in most applications). Random coordinate values are generated in each patch on $\mathbb{C}P^4$. All points lying within $\pm\epsilon$ of the Calabi-Yau hypersurface are kept and projected onto it orthogonally with respect to the Fubini-Study metric g^{FS}. The volume form μ is independent of $h_{A\bar{B}}$ and the points are calculated once at the start of the computation. In practice, $N_{points} = 3000$ is more than enough to get accurate results when minimising (compared to the 10^9 required by Donaldson algorithm). Generating the points can take from a few seconds to longer, depending on the value of ϵ. This is however an upfront cost as they don't need to be recomputed. Once the points are calculated, the values of $\mathcal{P}^l$, q^I and $\mathcal{Q}^l_{a\bar{b}}$, defined in eq. 31 are computed at each point. This is done with h_l initially equivalent to the Fubini-Study metric. These quantities are then used to compute $\Psi_{a\bar{b}}$ and v_ω. They form the initial values for the minimisation.

3.2.5 Summary of the set-up and minimisation

In summary, the algorithm has the following setps:

1. **Calculating bases**. For a given k, findthe basis of polynomials $\mathcal{P}^l$.

2. **Generating points**. Generate random points in $\mathbb{C}P^4$ with respect to the measure μ given in eq. 32 and project them onto X with the Fubini-Study metric. This yields N_{points} tuples of 4 homogeneous coordinates $\{u^\alpha\}$. This requires N_{points} and ϵ and outputs a list of 4 coordinate values, u^α, at every point.

3. **Calculating data**. Calculate the following quantities at every point and for each polynomial basis: Q_α, $\mathcal{Q}^l_{a\bar{b}}$. Construct the matrix $\Psi_{a\bar{b}}$ at every point. Putting these all together, compute the value of v_ω point-wise.

4. **Integrations**. Calculate V_ω and V_μ by performing a numerical integration. Construct the integrand together with the value of v_ω compute $H[\omega]$ through numerical integration.

5. **Minimisation**. Using the Levenberg-Marquardt method, minimise $H_1[\omega]$ to find the values of h_l closest to the Ricci-flat metric. The values of the components of h_l obtained fully characterise the Ricci-flat metric.

All functions required to this method are implemented in `Mathematica` and are available in the `Fermat.m` package at (37). Headrick and Nassar also wrote a `Mathematica` notebook, (that we extensively used) to illustrate their work and the use of the package. The notebook is available at (37) under the name `optimal.nb`. We encourage the reader to download the `Mathematica` package and notebook to generate results themselves and feel the satisfaction of computing a Calabi-Yau metric if they don't do it routinely.

3.2.6 Results

Headrick and Nassar apply the energy functional method to the Fermat Quintic (and other threefolds in the generalised Quintic family). They also generate metrics for the Quartic 2-fold. The reader is referred to their result section for a detailed discussion.

Note that the volume form μ, as presented in section 3.2.4 is not the one integrated over in practice. The latter, denote $\tilde{\mu}$ is a numerical approximation to μ.

They differ by a scaling factor which is irrelevant to the results since the functional is minimised for both. However, Headrick and Nassar work in their paper with normalised integrals. The following is explained in detail in section 4.5 of (40) and 6.2 of (1). They define the quantity

$$\eta_\omega = \frac{v_\omega}{\langle v_\omega \rangle}. \tag{33}$$

The expectation value of v_ω is defined through

$$\langle v_\omega \rangle = \frac{1}{V_{\tilde{\mu}}} \int_X \tilde{\mu} v_\omega \tag{34}$$

where $\tilde{\mu}$ is the approximate numerical measure and $V_{\tilde{\mu}}$ is defined in eq. 19. With η, the energy functional $H_1[\omega]$ defined in eq. 17 can be rewritten with correct normalisation as $E[\omega] = \left\langle (\eta_\omega - 1)^2 \right\rangle$. A quick manipulation shows that both functionals are equal up to a constant in the limit $\tilde{\mu} \to \mu$. In practice, the results are calculated with $H_1[\omega]$ and then normalised, as presented in (1).

The functional $E[\omega]$ is relevant as it is directly related to a standard measure of Ricci-flatness,

$$\sigma[\omega] = \int_X \tilde{\mu} |\eta - 1|. \tag{35}$$

This is used in particular to assess the quality of the metrics in (4; 3). Headrick and Nassar compute it for various k. They find that the value of σ decreases exponentially with k, which is a significant improvement compared to the polynomial decrease found in (3) using Donaldson's method. This effectively means that for a given k, metrics calculated with this method will be more Ricci-flat than Donaldson's balanced metrics. This is clearly shown in figs. 1 and 2 of (1) where they

plot the value of $E[\omega]$ as a function of h_l for the Quartic in the case of $k = 2, 3$ (where h_l is respectively 1 and 2 dimensional). They show on the figures the position of the balanced metric computed with Donaldson which doesn't correspond to the minimum of $E[\omega]$. In comparison, their own metrics visibly correspond to the minimum of the functional, thus again highlighting the improved accuracy. Overall the results confirm the robustness of the method by showing the obtained accuracy is comparable to existing methods while requiring fewer resources. In particular, it can produce highly accurate metrics on the conifold Quintic, a phenomenologically relevant space treated in (43).

While Headrick and Nassar have developed an elegant, easy to implement and efficient method to compute Calabi-Yau metrics of the Quintic and Quartic Calabi-Yaus, we remind the reader that generating similar results for other threefolds like the Schoen manifold used in (39) would require significantly more resources. Indeed, we wouldn't be able to reduce the basis $\mathcal{P}^l$ as we move away from the highly symmetric Fermat Quintic. This directly translates to a longer initial step to calculate the basis. However, the increased size of h_l also results in longer subsequent steps (apart from the generation of the points which is independent of the basis). This increase in time can be evaluated by measuring the time taken by each step detailed in section 3.2.5. This was done for dimensions of the basis varying between 2 and 460, corresponding to $k = 2$ to 19 in integer steps for the Quintic. This is shown in fig. 2.

The times shown in fig. 2 clearly demonstrate that the minimisation step increases fastest and thus becomes "problematic" first. This motives our attempt to replace the last step with a neural network. Indeed, using fig 9 in (1), we estimate that a non-reduced basis of dimension 1000 would correspond to $k = 3$ for a general

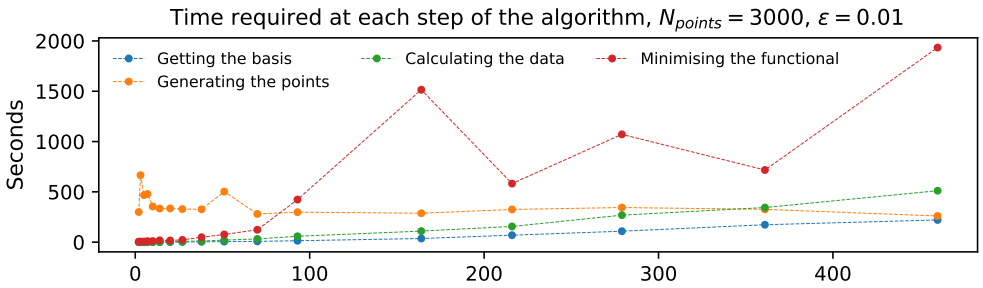

Figure 2: We show times (in seconds) taken to compute the steps described in section 3.2.5 for different basis dimensions. This demonstrates that the ultimate minimisation step scales the fastest with the dimension of the basis.

threefold. The time taken in that case isn't on the figure as the maximum dimension we managed to work with was 460. We expect that it would take order of hours, if not days (similarly to Donaldson's method) to compute metrics with 1000 dimensional bases. If we were to successfully replace the minimisation step by a faster method without losing too much accuracy, we could hope to compute metrics on more physically relevant spaces using a modified version of the energy functional method. We now investigate whether this is possible using a neural network.

4 Machine learning the Kähler potential

The results presented in the remaining sections are fully original, building up from the code developed by Headrick and Nassar available at (37). We take caution to highlight where their contribution ends and where our work starts.

The field of machine learning consists of the investigation, development and use of algorithms which automatically perform their task better as the number of times they've been executed increases. This ranges from "learning" chess moves, driving or classifying different objects (like the 10^6 samples of handwritten digits compiled in (44)). In practice, the list doesn't end and various forms of machine learning can

be applied to most problems. This wide range of possible uses motivated the development of various algorithms and diverse approaches to machine learning. Some of these are described in (10; 45), which are our main sources for what follows.

Three main methods of learning have been developed in particular. These are unsupervised, reinforcement and supervised learning. We say a few words about the first two and focus on the third, which we use in our implementation.

In unsupervised learning, algorithms are designed to group some input data without any guidance. Doing so, it highlights clusters, similarities or categories. This has been recently used on data sets of symptoms observed in COVID-19 patients. The analysis revealed 6 different distinct groups of symptoms and associated illness severity (46).[20]

Reinforcement learning is similar to unsupervised learning but requires a notion of quality of performance. This is particularly appropriate when there is a preferred output to the task at hand but no information regarding how to obtain it. In practice, the algorithm is rewarded when producing the preferred output. This is the approach taken for example in (47) to explore the type IIA string landscape. In particular, different string vacua are scanned over and the algorithm is rewarded proportionally to how close the corresponding physical properties are to the Standard Model.

Supervised learning is essentially a highly non-linear regression method. To train the algorithm, an ordered set of input (for example pictures of handwritten digits) and output (what the digits represent) is required. This however requires being

[20]While a reference to applications of unsupervised learning related to our field would have been appropriate here, we couldn't resist highlighting that it is 2020 and point the reader to an interesting topical study.

able to perform the task at hand in the first place. Once the algorithm is trained, it can be used to produce previously unknown output given more input. This is particularly useful in the case of difficult tasks that can't be accomplished more than a limited number of times. This approach is commonly used in theoretical physics, for example in (4; 18; 48; 49). It is also well suited to many problems in algebraic geometry, as highlighted in (10).

This is the approach we take in the present work using a neural network algorithm. Our input data are the necessary quantities to compute the functional calculated at every point and our output is ψ (the exponential of the Kähler potential) calculated at every point with the minimised h_l. Each random point on the manifold constitutes a training example. Once the algorithm is trained on a set of points, it can be used to calculate ψ at additional points, bypassing the computationally expensive integration and minimisation process originally used.

The process is split between the learning and the validation phases. During the learning phase, the algorithm is given input and output pairs (the training set) and adjusts its internal parameters in order to match the given outputs as well as possible. The larger the training set, the more the algorithm can fine-tune its parameters and the better it will perform. In order to measure how well it has learned, some data is kept unseen from the algorithm. This forms the validation set. Only the inputs only are given to it and the corresponding predicted values are compared against the known output. The fraction of the total known data used for training, $\gamma \in (0, 1)$, is typically varied. The algorithm is said to be learning if its performance on unseen data improves with γ.

Our input point-wise data is calculated with Headrick and Nassar's code. It contains

– the values of the coordinates: 4 complex numbers,

– the gradients Q_α: also 4 complex numbers,

– the matrix $\mathcal{Q}^l_{a\bar{b}}$ defined in eq. 31: 25 complex numbers for each basis element.

We work with $k = 10$, an intermediary providing a balance between accuracy of the results and computation time. This yields for the Quintic 38 basis elements corresponding 958 complex input values per point. The neural network is implemented using the `TensorFlow` library in `Python` (50). Consequently, inputs must be real and the complex numbers are separated into real and imaginary parts which equates 1916 real values for each point on the manifold. These 1916 are then given to the neural network. Their values, like an electrical signal, will propagate by activating different nodes going forward to produce an appropriate value of ψ. To differentiate the known and predicted values, we use the following convention. The values of ψ calculated with the energy functional minimisation method are labelled ψ and the associated values computed with the neural network $\hat{\psi}$.

4.1 Neural networks and their parameters

Now that we have introduced our training data, we define a neural network and its parameters in more detail. We present the specific network we have used in fig. 3 and give a flavour of how the network can "learn" the data. We focus on the conceptual perspective and attempt to keep the mathematical details to a minimum for conciseness and clarity. A clear introductory discussion can be found in the video series "Neural Networks" (51) on 3Blue1Brown by Grant Sanderson, a Stanford alumni in mathematics. The videos go into more detail regarding the adjustment

of parameters during training. For a more academic account, we direct the reader to (45) and (10), both accessible and tailored to theoretical physics applications.

A neural network is a network (i.e. points linked together by edges) which roughly imitates how the brain processes information. The network is built out of successive layers of nodes. Nodes in one layer aren't connected to each other but all of them are connected to the next layer. This is shown in fig. 3. In analogy with a brain, nodes are like neurons and the edges correspond to all possible synapses. Nodes each take a value x, corresponding to how active they are. The values of the nodes in the initial layer are the input data. The initial data is propagated forward through the edges from the input to the output layer, where we wish to find the value of ψ. Consequently, our first layer consists of 1916 nodes and our last one is a single node. Intermediate layers are called hidden layers and can be of any size. The network we have picked consists of 3 hidden layers with 150, 70, 10 nodes each. There is no hard rules about which network structure would give the best result for a given problem. Consequently, the structure presented here was found to work best through trial and error. The structure of the network is called a hyper-parameter: it can be varied to improve the overall performance but isn't adjusted during training.

We now describe the internal parameters of the network: weights (a) and biases (b) of each node and activation functions (f) of each layer impact and characterise the flow of information through the network. The weights and biases make up the internal parameters which are adjusted during training. The activation functions are hyper-parameters of the network. When discussing the following in the literature, the nodes' values, weights and biases are labelled with indices to specify their position in the network. Similarly, the activation functions carry an index linked to the layer they act on. For simplicity and clarity, we highlight how these parameters

play together to create a unique flow of information in the network but don't use any indices. [21] It is understood that the weights and biases take different values for each node and activation functions take different form depending on the layer. The set up and following discussion are illustrated in fig. 3.

Given a node with input, or activation value, x, the weight a and bias b act on x to give $y = ax + b$. This value y will be transmitted to all connected nodes in the next layer. When transmitted through an edge, this value y is modified by the activation function f associated with the layer. The following connected nodes then receive their own activation value, $x = f(y)$. In most cases, as shown in fig. 3, nodes in subsequent layers are connected to more than one node in the previous layer and therefore receive more than one $x = f(y)$. Then, their activation value x, which is turn modified by their parameters and transmitted further, is the sum of the activation values coming from all connected nodes in the previous layer. Through this process, the input is modified, transmitted through all nodes and edges (in our case $\simeq 300,000$ edges) and produces in the last layer a single output.

This network is essentially a function, that takes in 1916 arguments and maps them onto the reals. If all activation functions were set to the identity, the network would correspond to a linear function with $298,841$ parameters and the learning process would simply become a classic optimisation process over an extremely large parameter space. It would also likely not produce any "correct" output unless there was a way to write ψ as a linear function of the input data. The power of neural networks comes from the non-linearity of the activation functions. A list of most used activation functions together with their advantages and drawbacks can be found in (45). Again, there is no hard rule when choosing activation functions but one

[21] A more rigorous description can be found in (45).

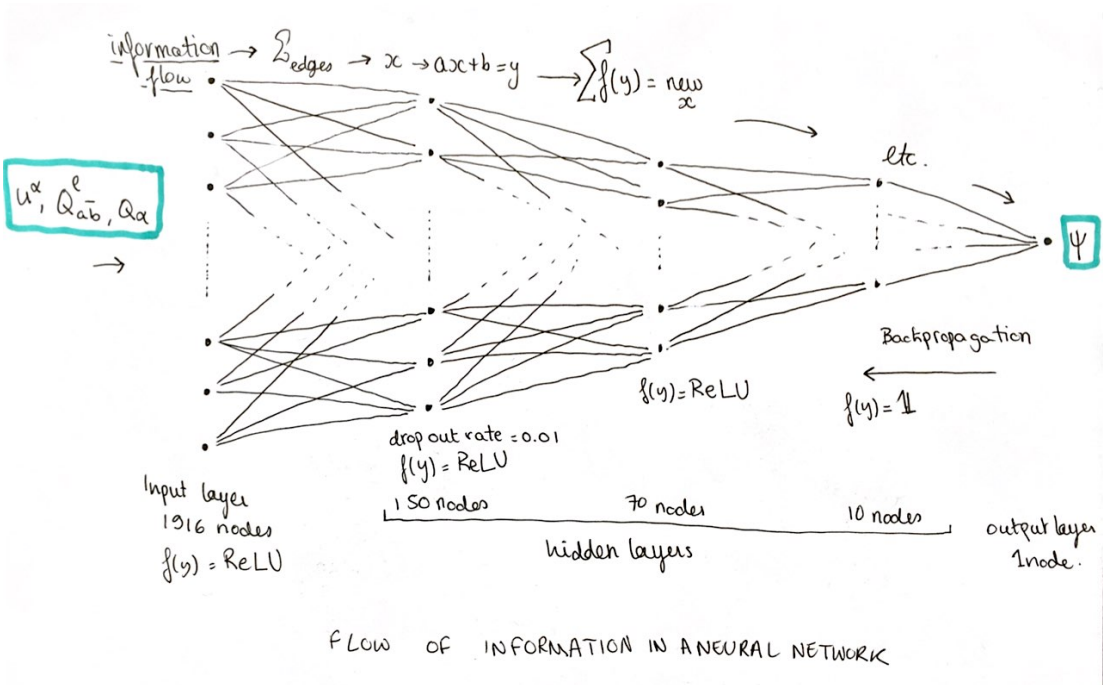

Figure 3: The specific structure of neural network we have implemented. The flow of information between the first and last layers is highlighted. We precise the hyper-parameter of the network.

should be mindful of the type of input data at hand as some activation functions are more suitable for particular data sets (depending on range, negativity of values, etc.).

In our case, we found the Rectified Linear Unit (ReLU) activation function to produce particularly accurate results. This was used for all layers but the last one where no activation function was implemented. By introducing non-linearity in the data flow, the network becomes able to reproduce highly complex, non-linear functions, like the underlying link between the input data and ψ which, as we saw in Section 3.2, corresponds to an integration over many points and the minimisation of a functional. This is particularly fascinating considering that, once trained, the network can produce ψ with knowledge of $\mathcal{Q}^l_{a\bar{b}}$ and Q_α at a single point.

Now that we have described how the network acts like a non-linear function, we sketch out how it "learns" during training and how to measure its progress during validation. As we shall see in more detail, this is simply an optimisation process where the internal parameters of the network are adjusted during a so-called back-propagation. In practice, the parameters are adjusted to minimise a loss function, akin to the sum of squares used in most linear fits. What is described now happens automatically during the training:

- The network is given a batch of random training elements from the training set. The network runs each input and collects the output it predicts with its current parameters for each element.

- Each computed output is compared to the known output and the corresponding value of the loss is calculated for all the points.

- Going backwards from the last to the first layer, all parameters are adjusted to minimise the loss for all the points in the batch. This is back-propagation.

- New batches are given to the network until all the data has been seen.

This process is called an epoch. The network can see the same training data multiple times throughout an epoch but this isn't a problem as the combinations of points in the batches is unlikely to repeat. In practice, the network is trained over multiple epochs, improving its parameters further every time. In our case, we train during 100 epochs and with 70 points in each batch.

The loss function we minimise is the mean absolute percentage error (MAPE) given by

$$MAPE = 100 * \frac{1}{n_{batch}} \sum_{i=1}^{n_{batch}} \frac{|\psi - \hat{\psi}|}{\psi} \tag{36}$$

where n_{batch} is the number of points in a batch, which is 70 in our case. At each epoch, the value of the MAPE is calculated for both the training and validation data sets, giving us an overview of the training progress. In the case of the validation data set, the MAPE isn't computed over the points in a batch but over all points in the validation set.

Before we present the results obtained with the neural network described above, we introduce a common pitfall of machine learning: over-fitting. This happens when the performance of the model improves as it sees more training data but stagnates (or worsens) when tested on validation data. In a standard fitting procedure, this corresponds to using a fit function with more parameters than the number of data points fitted. In the case of neural networks, this is also associated to networks whose structures are overly complex. An easy way to minimise over-fitting is therefore to keep the network as small and simple as possible, while still ensuring good performance. Over-fitting also means that the network is "learning" local structures in the training data (and therefore improves) but this doesn't translate to better performance on unseen data. A way to mitigate this effect is to implement a so called drop-out rate for some of the layers. A proportion of the nodes are then randomly set to zero (i.e. turned off) during each pass over the data. The set of deleted nodes are randomly picked each time which forces the network to learn more global structures and effects in the data. In our computation we found that a drop-out rate of 0.01 in the first hidden layer was sufficient to avoid over-fitting.

4.2 Performance of the network during training

The network described above is implemented to machine learn data calculated with Headrick and Nassar's `Mathematica` package available at (37). As discussed above,

the `Fermat.m` package is used to calculate the coordinates, values of Q_α and $\mathcal{Q}^l_{a\bar{b}}$ (collectively referred to as input data) as well as the point-wise value of ψ (the output data). This is done at 5000 points on the manifold and at $k = 10$. On the author's personal laptop, the whole process takes 16 minutes of which 10 are needed to generate the input data and 6 to perform the minimisation and obtain the values of ψ.

4.2.1 A single training

We train the network once with $\gamma = 0.7$, meaning that 3500 points are used for training and 2500 remain unseen and are used to validate the data. After each epoch, the values predicted are compared to the known output by calculating the loss function defined in eq. 36. This is done for the points that the network was trained on as well as for the remaining validation points.[22] Both are plotted against the number of epochs in fig. 4. The network is trained over batches of 70 points during 100 epochs. The value of the loss decreases during training as shown in fig. 4. The network is learning the underlying features primarily during the first 40 epochs, after which it predicts output with a mean error below 3.5%. During the remaining 60 epochs, the discrepancy between the predictions and the known values further drops to 1.6% after 100 epochs. Continuing training after 40 epochs often results in over-fitting where the value of the loss on the training set decreases while the performance on the validation data stagnates. As explained in the previous section, this was prevented by using drop-out in the first hidden layer.

At first sight, fig. 4 visually shows no performance gap between the training and validation data sets. A precise look at the values tells a different story. In fact, the

[22]There is no back-propagation on the validation data and the network doesn't learn on it. They therefore remain "unseen" during the entire training.

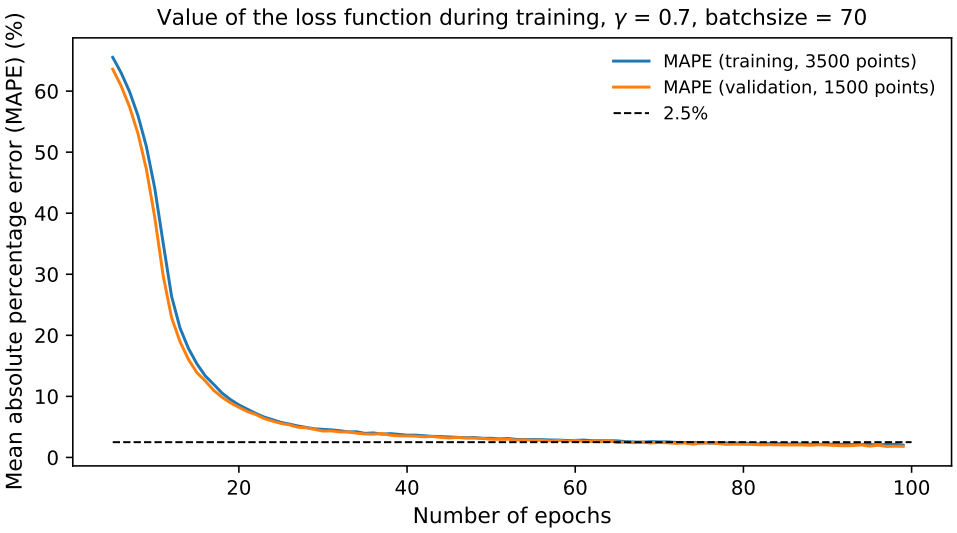

Figure 4: The values of the loss for both the training and validation sets are plotted after each epoch. All 100 epochs constitute the training. For indication, the 2.5% error mark is superimposed.

value of the loss on the training set is 1% above that on the validation set. This is a direct effect of the implemented drop out. When nodes are randomly removed (during training), the network doesn't perform as well as when all nodes are used (during validation). This discrepancy can appear surprising when recalling the drop-out rate is 1%. Consequently only 1.5 nodes on average are removed during each pass. This highlights the subtleness of neural networks: a single removed node can significantly alter the performance.

This is a good time to recall that the network, once trained, acts essentially as a fit function. Tools of statistical analysis and modelling can therefore be used to assess its performance further. We generate $10,000$ additional inputs and outputs using Headrick and Nassar's code.[23] The trained network (with the parameters described above) is now used to predict the values of ψ for the $10,000$ points (denoted $\hat{\psi}_i$

[23]None of these points are used during training or validation and they are all completely unseen.

where i labels the points). This takes a single second! The $\hat{\psi}_i$ are compared to the corresponding ψ_i found by minimising the functional (which have remained unseen by the network). This is shown in fig. 5 where $\hat{\psi}_i$ is plotted against ψ_i. Ideal

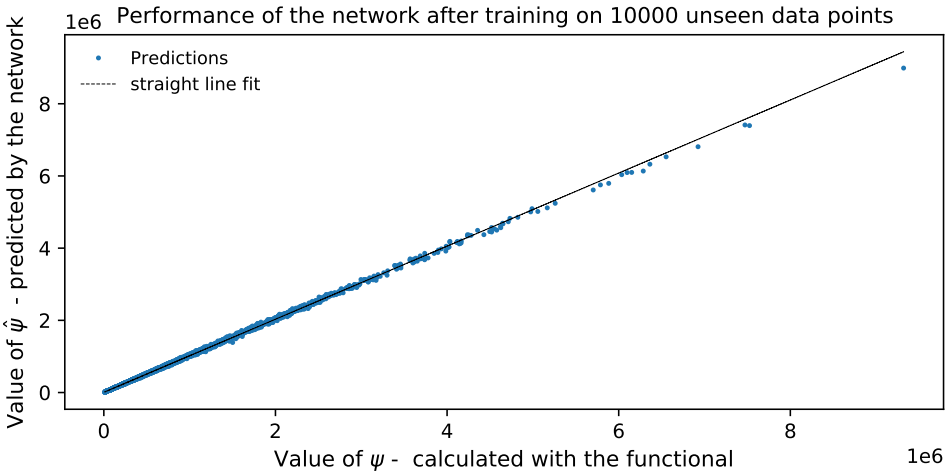

Figure 5: The point-wise predictions of the trained network ($\hat{\psi}_i$) are plotted against the values calculated by minimising the functional (ψ_i) for $10,000$ unseen points. The result of a fit to 37 is superimposed. The optimised coefficients are $m = 1.012 \pm 0.002$ and $v = 6354 \pm 196$ where the uncertainties represent standard deviation errors on the parameters.

predictions by the network would results in the straight line $\hat{\psi}_i = \psi_i$. This can be visually observed in fig. 5. To gain a quantitative understanding, we fit the data shown in fig. 5 to the function

$$\hat{\psi}_i = m\psi_i + v \tag{37}$$

to obtain coefficients m and c. The ideal network would then correspond to fit coefficients $m = 1$ and $v = 0$. When fitting the data, we find that $m = 1.012 \pm 0.002$ and $v = 6354 \pm 196$ where the uncertainties are standard deviation errors on the parameters returned by the fitting algorithm. The value of m with its low uncertainty confirms the quality of the predictions. However, the significant offset found in v suggests that the network tends to systematically underestimate slightly

the value of ψ. The mean value of ψ over the $10,000$ points lies at $433,971$. The offset then corresponds to an average systematic shift of 1%. This is of the same order as the 1.6% MAPE error observed after training in fig. 4. This systematic shift could potentially limit the minimum reachable error. Improving this discrepancy would require changes in the hyper-parameters. In particular, we notice in fig. 5 that the network performs worst for larger values of ψ. Using different activation functions could potentially resolve this.

4.2.2 Training over multiple γ

To assess the overall performance of the network, the proportion (γ) of data given to the network for training is varied between 0.01 and 1 in steps of 0.02. For each γ in the range, the network is trained from scratch using batches of 70 points over 100 epochs. After training, the network is given the remaining unseen points which constitute the validation set. One then computes the overall percentage error in the output from the unseen data for each γ. The plot of the mean absolute percentage error (MAPE) against γ is commonly called the learning curve of the network.

In order to measure the uncertainty in the predictions, the process described above is repeated 10 times using the same set of 5000 points (which are redistributed between the training and validation set every time). An average of the MAPE value at each γ is taken over the 10 runs and the corresponding standard error on the mean is computed. Both are plotted against γ in fig. 6. It can immediately be seen that the mean value of the loss (itself a mean percentage error) decreases as the amount of data seen during training increases. This confirms that the algorithm is learning from the data given. Additionally, the error on the mean MAPE decrease as γ increases, showing the network is learning reliably. When learning from $\gamma = 0.4$

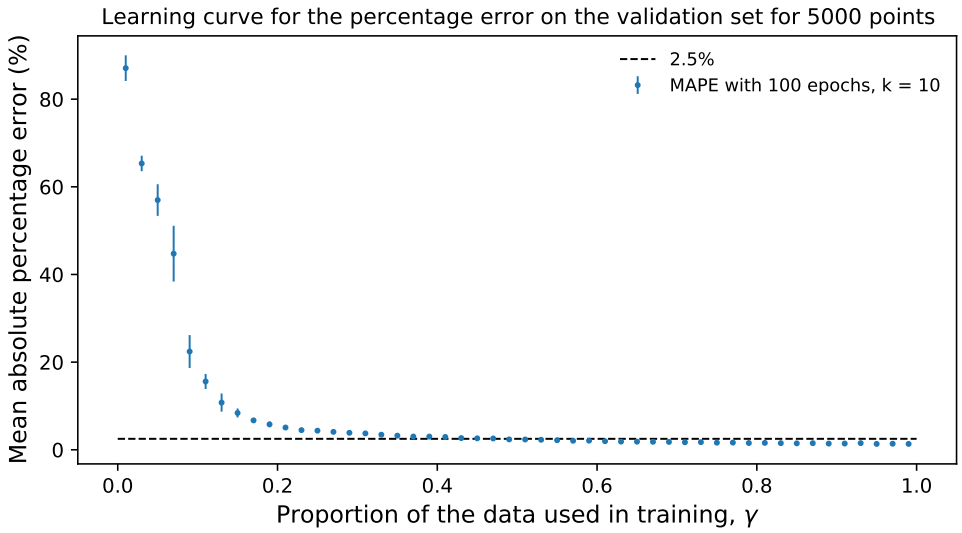

Figure 6: The value of the mean percentage error (eq. 36), averaged over 10 runs is shown at each γ. The errorbars represent the standard error on the average. We superimpose the 2.5% error threshold. The neural network trained using 5000 points

of the data (i.e. 2000 points), the MAPE is already below 3%. It reaches $2.5\% \pm 0.1$ at $\gamma = 0.46$ (i.e. having seen 2300 points). For $\gamma = 0.78$ (3900 points), the error drops to $1.5\% \pm 0.07$. While extremely promising, we emphasize that this doesn't say anything about the Ricci-flatness of the results. Hence we can't compare right away to results obtained through different methods presented in (4; 1). However, the values of the MAPE obtained during training shown in fig. 6 do demonstrate that the simple network structure we have constructed is able to learn the features of the data from Headrick and Nassar's code to a high accuracy.

To confirm this, we computed the coefficient of determination R^2,

$$R^2(\psi, \hat{\psi}) = 1 - \frac{\sum_i (\psi_i - \hat{\psi}_i)^2}{\sum_i (\psi_i - \bar{\psi})^2} \tag{38}$$

where $\bar{\psi}$ is the statistical mean of all ψ_i (10). This coefficient measures the degree of

linearity between the ψ_i's and $\hat{\psi}_i$'s. It becomes 1 if both are identical (corresponding to a perfect fit). The value of R^2 is positive if there is correlation between $\hat{\psi}$ and ψ. In particular, if the network always produced the constant average of the ψ_i, we would find $R^2 = 0$. No correlation results in a negative R^2. This is calculated for the validation data after each epoch during training. We plot an average of the 10 runs with the corresponding standard error on the mean in fig. 7. The R^2 coefficient

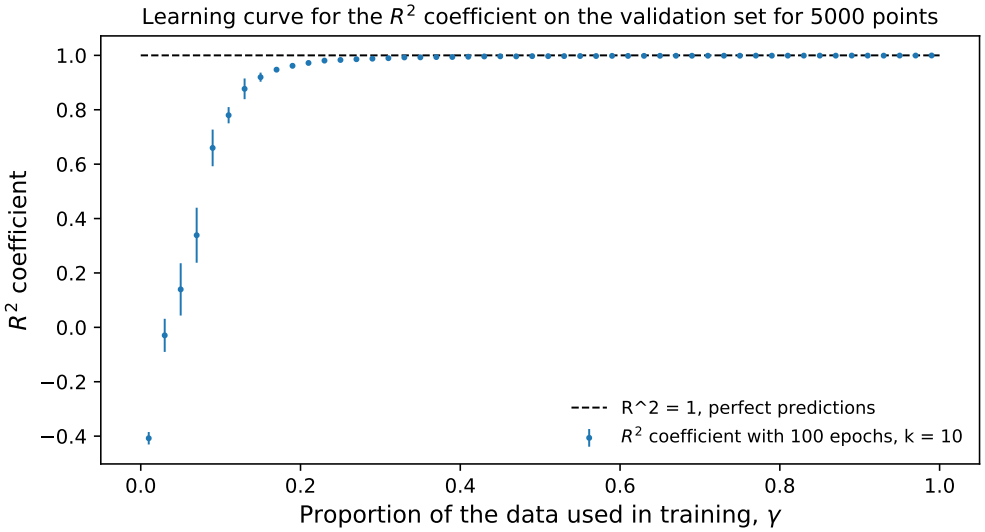

Figure 7: The value of the R^2 coefficient averaged over 10 runs is shown at each γ. The error bars represent the standard error on the mean. We superimpose $R^2 = 1$ representing a perfect fit. The neural network is trained using 5000 points

visually reaches 1 around $\gamma = 0.4$, thus confirming that 2000 points are enough for the network to learn the underlying features of the data. The exact values found are $R^2 = 0.996 \pm 0.001$ for $\gamma = 0.46$ and $R^2 = 0.998 \pm 0.0003$ for $\gamma = 0.78$.

Overall, this demonstrates that the network is capable of learning with great accuracy the features underlying the relationship between ψ and the corresponding input data. In particular, it is capable of predicting ψ with less than 2.5% error after having seen only 2300 data points. Training the network on 3500 points (with

error around 1.5%) only takes 2 minutes and obtaining the predictions on $10,000$ additional points once the network is trained takes under 2 seconds. This may seem like a considerable gain from the 6 minutes required to compute the 5000 values of ψ with the minimisation. However, we shouldn't forget that, to obtain the training data necessary to reach 1.5% error, 3500 points need to be computed using the energy functional. This itself takes 12 minutes of which 3.5 are spent on the minimisation. Now we also recall that, with 3000 points, the energy functional method generates highly Ricci-flat metrics. This yields the overall coefficients h_l which are used to evaluate ψ and consequently the Kaähler potential. However, ψ can be evaluated using h_l on as many points as we have coordinates for, beyond the ones used to obtain the coefficients.

If the problem was to compute ψ on more points, our neural network would be an efficient solution. However, once we have obtained the training data, there is no barrier to computing the Kähler potential on additional points using the existing method in (1). Estimating the time gained by replacing the minimisation by our network isn't so obvious anymore. We explore in the next section whether our network can provide a significant improvement to the energy functional method. Going back to our ultimate aim, we then try to conclude on the to viably of our hybrid method to predict numerical Calabi-Yau metrics on manifolds beyond the Quintic.

4.3 Using our network for research

An important feature of figs. 6 and 7 is the decrease of the error on the mean of the MAPE and R^2 coefficients. The values of the MAPE and R^2 themselves already show that the quality of the predictions of the network is poor for $\gamma \leq 0.2$. This

is confirmed when looking at the size of the error bars on both figs. 6 and 7. The deviation are largest for $\gamma \leq 0.2$ meaning that the network's outputs vary in accuracy from run to run and thus produce unreliable outputs. However, this stabilises for $\gamma \geq 0.2$ which corresponds to the minimum amount of training data required for the network to effectively learn. The deviation in the quality of the predictions decreases significantly and the network produces values with an accuracy that can be trusted. This feature is essential as it shows that the network would produce values of ψ with a close-to constant accuracy. Consequently, if it was used to generate values of ψ for which we don't have the value calculated using the energy functional, we would know to trust these to 2.5% and 1.5% if the network was trained respectively on 2900 or 3900 points. This essentially shows that a neural network is capable of quickly learning (2 minutes) and then predicting (1 second) trustworthy values of ψ with $\sim 1.5\%$ error. This means that the simple structure detailed in section 4.1 is capable of emulating the relationship between ψ, the exponential of the Kähler potential at a point and the point's coordinates and values of Q_α and $\mathcal{Q}^l_{a\bar{b}}$.

We now attempt to resolve whether our hybrid method using a neural network coupled to the energy functional method by Headrick and Nassar could offer a viable alternative to currents methods used to compute numerical Calabi-Yau metrics, in particular (1; 3; 4). This decomposes into two questions: are the metrics we produced with our method as Ricci-flats as the ones from (1; 3; 4) and if so, would our method offer a significant speed up to (1). The former couldn't be answered here due to lack of time. It would require computing the value of the functional σ defined in eq. 35. This measure of Ricci-flatness is commonly used in the literature and obtaining it for the metrics produced with the neural network would quickly show how their Ricci-flatness compares. This remains one of the main investigations necessary to

evaluate the feasibility of our method.

Whether our method provides a significant time gain has already been discussed briefly in the previous section. We remind the reader that as we consider more sophisticated spaces, the basis of polynomials can't be reduced using symmetry arguments anymore. The dimension of the space minimised over thus increases greatly. As shown in fig. 2, the time required to minimise the functional is likely to scale faster than the time needed to compute the input data. This shows that a small gain of time on the Quintic would become more significant when applying our method to more sophisticated manifolds, our ultimate motive. Using our network in that case however still requires producing data using the energy-functional method, including its ultimate minimisation step. As we have demonstrated in section 3.2.6, this appears severely limited by the time required to carry out the process. As it stands, our network doesn't appear to be a viable solution to the drawbacks of (1). It is unlikely that our network also produces metrics of better accuracy (in the sense of Ricci-flatness) than the ones already calculated for the training set. There are, however, slight modifications that could make the method competitive.

An example that comes to mind is to use as a loss function a measure of Ricci-flatness directly similarly to (49). One could then imagine calculating approximately Ricci-flat metrics (for example using few points and/or lower k) with the minimisation method. This would provide a low accuracy training set quickly and thus bypass the time limitations pointed out above by compromising on the accuracy. Using a similar algorithm as (4), we can then use our neural network, with an adapted loss function, to produce Calabi-Yau metrics closer to Ricci-flat than originally computed for the training set. Once trained, it could be used to predict the Kähler potential on a larger sample of points. Such a procedure could offer a significant gain of time

and become a viable solution to the limitations of the energy functional method. In an alternative direction, we could ask whether a network trained on data from the Quintic could successfully predict the output of a different Calabi-Yau. This would require generating Quintic training data using Headrick and Nassar's method. Consequently, the minimisation step for the other Calabi-Yau could be fully bypassed, thus providing a significant speed-up.

5 Conclusion

In this thesis, we studied numerical Calabi-Yau metrics and showed through original computations that machine learning can provide significant short-cuts.

To this end, we have introduced concepts of complex geometry needed to understand Calabi-Yau manifolds and the methods developed to obtain Ricci-flat metrics on them. We have also provided an introduction to machine learning, in particular supervised learning with neural networks. Moreover, we reviewed different methods of obtaining Ricci-flat metrics, emphasizing the energy functional approach. We then showed that a simple neural network was able to predict Kähler potential on the Quintic to high precision. Finally, we discussed whether our method could be used to compute metrics on physically relevant compactification threefolds.

Finding the Ricci-flat metric was motivated by string theory as it is one of the main ingredients required to obtain further predictions on existing minimally supersymmetric standard models (MSSM). Such predictions are in turn necessary to reject or further confirm some of these models. As it stands, the energy functional method can't be used to evaluate metrics on compactification spaces beyond the Quintic. The main limitation is the time required to obtain results on spaces where

the symmetries can't be exploited. This is the case of most relevant Calabi-Yaus.

However, if the method were to take less time, it could become viable for other threefolds. In that direction, we attempted to replace the ultimate integration and minimisation step of the energy functional method by a neural network. We have shown that a (relatively) simple network is capable of learning in a few minutes how to predict the Kähler potential on the Quintic Calabi-Yau. The obtained values were found to have 1.5% average discrepancy from values calculated using the energy functional method developed in (1). This is in itself quite an impressive achievement considering the mathematical complexity of the underlying data.

To be able to answer whether our hybrid method could offer a viable alternative to existing methods would firstly require an assessment of the Ricci-flatness of the resulting Kähler potential. If our results are of comparable accuracy to those of the energy functional method, it still remains to consider whether our network implementation does provide a speed-up of the original method. As we have discussed, this isn't an easy question. However, slight changes to the method, e.g by using a measure of Ricci-flatness directly as a loss function, could potentially provide a significant improvement and lead to an alternative method of calculating numerical Calabi-Yau metrics. We hope to investigate this line of research further and seek answers to the remaining questions.

Acknowledgments

Writing this dissertation and consequently finishing my MSc in the time of quarantines, viruses and isolation was certainly a special experience. I would like to first and foremost thank the people who loved me, supported me and carried me through the darkest days. I remain unsure where I would have found the strength to continue if they hadn't stood by me this year. For this, I thank my partner, Adam Chalabi, my flatmates Zohar Mendzelevski-Steinberg, Cerian Healey and Shree Thirumalaikumar as well as my "distanced" friends who I felt near me through lockdown and the summer even though I haven't seen them in so long. I'm thinking especially of Cyrus Skeete and his wonderful optimism and reassurance. I also would like to thank my parents, Pierre and Marie Calmon for their support this year, emotional and financial as well as Reema Chalabi, who cared for me like a daughter during lockdown and whose presence was particularly reassuring in the first few weeks of this project when it all felt so daunting.

Working remotely for exams was extremely difficult and I would like to thank in particular Dan Waldram. His presence, in particular during our Friday morning class calls has been a source of support and reassurance. I am grateful for his efforts to keep the class' spirit going throughout this difficult summer and hope he is aware of how much of a difference he made.

Last but not least, I am extremely grateful to Yang-Hui He for accepting to supervise me through this project. I had been excited to work with him since I met him in February and I enjoyed discovering his field and passion for machine learning through its potential for string theory compactification. This project was his idea. It felt like the right challenge and a satisfying way to end my taught education. I would also like to warmly thank Anthony Ashmore, who took the time to help me understand most of the concepts in the thesis and supported me throughout the project. I wouldn't have been able to go this far with the project if it wasn't for his patience. I am also extremely grateful for his comments

on this manuscript that he could provide on such short notice.

While I am disappointed I had to work isolated on this and have missed the feeling of belonging to a group, I particularly enjoyed Yang's upbeat spirit and Anthony's kindness and hope there will be more opportunity for all of us to work together more closely in the future.

Finally, I thank Hunee for his tunes which accompanied me throughout the project and kept my spirits up when the machine learning was really not learning.

References

[1] M. Headrick and A. Nassar, *Energy functionals for Calabi-Yau metrics*, *J. Phys. Conf. Ser.* **462** (2013) 012019.

[2] S. Donaldson, *Some numerical results in complex differential geometry*, *Pure and Applied Mathematics Quarterly* **5** (01, 2006) .

[3] M. R. Douglas, R. L. Karp, S. Lukic and R. Reinbacher, *Numerical Calabi-Yau metrics*, *J. Math. Phys.* **49** (2008) 032302, [hep-th/0612075].

[4] A. Ashmore, Y.-H. He and B. A. Ovrut, *Machine learning Calabi-Yau metrics*, 1910.08605.

[5] P. Candelas, G. T. Horowitz, A. Strominger and E. Witten, *Vacuum Configurations for Superstrings*, *Nucl. Phys. B* **258** (1985) 46–74.

[6] G. Tian and S.-T. Yau, *Three dimensional algebraic manifolds with* $c_1 = 0$ *and* $\chi = 6$, *Conf. Proc. C* **8607214** (1986) 543–559.

[7] B. R. Greene, K. H. Kirklin, P. J. Miron and G. G. Ross, *A Three Generation Superstring Model. 1. Compactification and Discrete Symmetries*, *Nucl. Phys. B* **278** (1986) 667–693.

[8] B. R. Greene, K. H. Kirklin, P. J. Miron and G. G. Ross, *A Three Generation Superstring Model. 2. Symmetry Breaking and the Low-Energy Theory*, *Nucl. Phys. B* **292** (1987) 606–652.

[9] M. Kreuzer and H. Skarke, *Complete classification of reflexive polyhedra in four-dimensions*, *Adv. Theor. Math. Phys.* **4** (2002) 1209–1230, [hep-th/0002240].

[10] Y.-H. He, *The Calabi-Yau Landscape: from Geometry, to Physics, to Machine-Learning*, 1812.02893.

[11] M. Reid, *The moduli space of 3-folds with k=0 may nevertheless be irreducible*, *Mathematische Annalen* **278** (1987) 329–334.

[12] S. Ashok and M. R. Douglas, *Counting flux vacua*, *JHEP* **01** (2004) 060, [hep-th/0307049].

[13] Y.-H. He, *Machine-learning the string landscape*, *Phys. Lett. B* **774** (2017) 564–568.

[14] D. Krefl and R.-K. Seong, *Machine Learning of Calabi-Yau Volumes*, *Phys. Rev. D* **96** (2017) 066014, [1706.03346].

[15] J. Carifio, J. Halverson, D. Krioukov and B. D. Nelson, *Machine Learning in the String Landscape*, *JHEP* **09** (2017) 157, [1707.00655].

[16] F. Ruehle, *Evolving neural networks with genetic algorithms to study the String Landscape*, *JHEP* **08** (2017) 038, [1706.07024].

[17] Y.-H. He, *Deep-Learning the Landscape*, 1706.02714.

[18] J. Bao, Y.-H. He, E. Hirst and S. Pietromonaco, *Lectures on the Calabi-Yau Landscape*, 2001.01212.

[19] B. A. Ovrut, A. Purves and S. Spinner, *The minimal SUSY BL model: from the unification scale to the LHC*, *JHEP* **06** (2015) 182, [1503.01473].

[20] V. Braun, Y.-H. He, B. A. Ovrut and T. Pantev, *The Exact MSSM spectrum from string theory*, *JHEP* **05** (2006) 043, [hep-th/0512177].

[21] A. Strominger, *Yukawa couplings in superstring compactification*, *Phys. Rev. Lett.* **55** (1985) 2547.

[22] M. R. Douglas, *Calabi–Yau metrics and string compactification*, *Nucl. Phys. B* **898** (2015) 667–674, [1503.02899].

[23] M. Headrick and T. Wiseman, *Numerical Ricci-flat metrics on K3*, *Class. Quant. Grav.* **22** (2005) 4931–4960, [hep-th/0506129].

[24] C. Doran, M. Headrick, C. P. Herzog, J. Kantor and T. Wiseman, *Numerical Kahler-Einstein metric on the third del Pezzo*, *Commun. Math. Phys.* **282** (2008) 357–393, [hep-th/0703057].

[25] V. Braun, T. Brelidze, M. R. Douglas and B. A. Ovrut, *Calabi-Yau Metrics for Quotients and Complete Intersections*, *JHEP* **05** (2008) 080, [0712.3563].

[26] L. B. Anderson, V. Braun and B. A. Ovrut, *Numerical Hermitian Yang-Mills Connections and Kahler Cone Substructure*, *JHEP* **01** (2012) 014, [1103.3041].

[27] V. Braun, T. Brelidze, M. R. Douglas and B. A. Ovrut, *Eigenvalues and Eigenfunctions of the Scalar Laplace Operator on Calabi-Yau Manifolds*, *JHEP* **07** (2008) 120, [0805.3689].

[28] M. Nakahara, *Geometry, topology and physics*. 1990.

[29] C. Hull, "Differential Geometry 2018-19." https://imperialcollegelondon.app.box.com/s/15jzssk41ak0ixxky70ry2j16rsmd590.

[30] E. Calabi, *On kähler manifolds with vanishing canonical class*, in *Algebraic geometry and topology. A symposium in honor of S. Lefschetz*, vol. 12, pp. 78–89, 1957.

[31] S.-T. Yau, *Calabi's Conjecture and some new results in algebraic geometry*, *Proc. Nat. Acad. Sci.* **74** (1977) 1798–1799.

[32] S. T. Yau, *On the Ricci curvature of a compact Kähler manifold and the complex Monge-Ampère equation.I.*, *Commun. Pure Appl. Math.* **31** (1978) 339–411.

[33] P. Green and T. Hubsch, *Calabi-yau Manifolds as Complete Intersections in Products of Complex Projective Spaces*, *Commun. Math. Phys.* **109** (1987) 99.

[34] T. Hubsch, *Calabi-yau Manifolds: Motivations and Constructions*, *Commun. Math. Phys.* **108** (1987) 291.

[35] P. Candelas, A. Dale, C. Lütken and R. Schimmrigk, *Complete intersection calabi-yau manifolds*, *Nuclear Physics B* **298** (1988) 493 – 525.

[36] S.-T. Yau, *A survey of calabi-yau manifolds*, *Surveys in differential geometry* **13** (2008) 277–318.

[37] M. Headrick and A. Nassar, "Mathematica packages by matthew headrick." `http://people.brandeis.edu/~headrick/Mathematica/index.html`.

[38] G. Tian, *On a set of polarized kähler metrics on algebraic manifolds*, *J. Differential Geom.* **32** (1990) 99–130.

[39] V. Braun, Y.-H. He, B. A. Ovrut and T. Pantev, *A Heterotic standard model*, *Phys. Lett. B* **618** (2005) 252–258, [`hep-th/0501070`].

[40] M. Headrick, "Our strategy." `http://people.brandeis.edu/~headrick/Mathematica/strategy.pdf`, 2009.

[41] W. R. Inc., "Mathematica, Version 12.1." `https://www.wolfram.com/mathematica`.

[42] M. Headrick, "Compendium of useful formulas." `http://people.brandeis.edu/~headrick/HeadrickCompendium.pdf`, 2020.

[43] D. Baumann, A. Dymarsky, S. Kachru, I. R. Klebanov and L. McAllister, *D3-brane Potentials from Fluxes in AdS/CFT*, *JHEP* **06** (2010) 072, [`1001.5028`].

[44] "The mnist database of handwritten digits." `http://yann.lecun.com/exdb/mnist/`.

[45] F. Ruehle, *Data science applications to string theory*, *Phys. Rept.* **839** (2020) 1–117.

[46] C. H. Sudre, K. Lee, M. Ni Lochlainn, T. Varsavsky, B. Murray, M. S. Graham et al., *Symptom clusters in covid19: A potential clinical prediction tool from the covid symptom study app*, *medRxiv* (2020) , [https://www.medrxiv.org/content/early/2020/06/16/2020.06.12.20129056.full.pdf].

[47] J. Halverson, B. Nelson and F. Ruehle, *Branes with Brains: Exploring String Vacua with Deep Reinforcement Learning*, *JHEP* **06** (2019) 003, [1903.11616].

[48] W.-C. Gan and F.-W. Shu, *Holography as deep learning*, *Int. J. Mod. Phys. D* **26** (2017) 1743020, [1705.05750].

[49] I. M. Comsa, M. Firsching and T. Fischbacher, *SO(8) Supergravity and the Magic of Machine Learning*, *JHEP* **08** (2019) 057, [1906.00207].

[50] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro et al., *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015.

[51] G. Sanderson, "Neural networks." https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi, aug, 2018.