

# JAXVacua — a framework for sampling string vacua

A. Dubey,<sup>a</sup> S. Krippendorf<sup>a,b</sup> and A. Schachner<sup>c</sup>

<sup>a</sup>*Arnold Sommerfeld Center for Theoretical Physics, Ludwig-Maximilians Universität, Theresienstr. 37, 80333 München, Germany*

<sup>b</sup>*Universitäts-Sternwarte, Fakultät für Physik, Ludwig-Maximilians Universität, Scheinerstr. 1, 81679 München, Germany*

<sup>c</sup>*Department of Physics, Cornell University, Ithaca, NY 14853, U.S.A.*

*E-mail:* [Abhishek.Dubey@physik.uni-muenchen.de](mailto:Abhishek.Dubey@physik.uni-muenchen.de),  
[sven.krippendorf@physik.lmu.de](mailto:sven.krippendorf@physik.lmu.de), [as3475@cornell.edu](mailto:as3475@cornell.edu)

**ABSTRACT:** Moduli stabilisation in string compactifications with many light scalars remains a major blind-spot in the string landscape. In these regimes, analytic methods cease to work for generic choices of UV parameters which is why numerical techniques have to be exploited. In this paper, we implement algorithms based on JAX, heavily utilising automatic differentiation, just-in-time compilation and parallelisation features, to efficiently construct string vacua. This implementation provides a golden opportunity to efficiently analyse large unexplored regions of the string landscape. As a first example, we apply our techniques to the search of Type IIB flux vacua in Calabi-Yau orientifold compactifications. We argue that our methods only scale mildly with the Hodge numbers making exhaustive studies of low energy effective field theories with  $\mathcal{O}(100)$  scalar fields feasible. Using small computing resources, we are able to construct  $\mathcal{O}(10^6)$  flux vacua per geometry with  $h^{1,2} \geq 2$ , vastly out-performing previous systematic searches. In particular, we showcase the efficiency of our methods by presenting generic vacua with fluxes below the tadpole constraint set by the orientifold with up to  $h^{1,2} = 25$  complex structure moduli.

**KEYWORDS:** Flux Compactifications, Superstring Vacua, String and Brane Phenomenology

ARXIV EPRINT: [2306.06160](https://arxiv.org/abs/2306.06160)

---

## Contents

|          |                                                                   |           |
|----------|-------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                               | <b>1</b>  |
| <b>2</b> | <b>Type IIB flux compactifications</b>                            | <b>4</b>  |
| 2.1      | Type IIB flux vacua                                               | 4         |
| 2.2      | Explicit models at large complex structure                        | 5         |
| 2.3      | Model construction and data-sets                                  | 7         |
| <b>3</b> | <b>Algorithmic approach for string vacua</b>                      | <b>8</b>  |
| 3.1      | EFT module — moduli potentials in JAX                             | 9         |
| 3.2      | Optimisation module — numerical search for extremal points        | 11        |
| 3.3      | Sampling module — sampling of fluxes and initial guesses          | 12        |
| 3.4      | Filter module — identification of trustworthy string vacua        | 14        |
| <b>4</b> | <b>Numerical experiments</b>                                      | <b>15</b> |
| 4.1      | Sampling procedures in practice — $h^{1,2} = 2$                   | 15        |
| 4.2      | Generating samples of flux vacua — $h^{1,2} = 4, 5$               | 18        |
| 4.3      | Scaling behaviour for large numbers of moduli — $h^{1,2} \leq 25$ | 19        |
| <b>5</b> | <b>Conclusions</b>                                                | <b>22</b> |
| <b>A</b> | <b>Implementation details of JAXVacua</b>                         | <b>23</b> |

---

## 1 Introduction

The string landscape is believed to contain a large number of vacuum solutions, current estimates ranging between  $10^{500}$  [1, 2] in type IIB flux compactifications to  $10^{272,000}$  [3] in F-theory constructions.<sup>1</sup> Despite these statistical estimates, it is fair to say that our capabilities to efficiently construct string vacua are extremely limited. To understand the low-energy physics attainable from string compactifications, a comprehensive numerical approach for analysing string theory vacua is therefore highly desirable. This is the overarching objective of this work. Success in this endeavour would enable an unprecedented look into the nature of consistent theories of quantum gravity and, in particular, their distinguishing features compared to bottom-up Effective Field Theories (EFTs).

More specifically, high-performance methods to search for realistic vacua will allow us to explicitly scrutinise existing constructions of e.g. string inflation [5, 6], while at the same time explore previously unknown mechanisms in the landscape. In this context, there are several computational steps which have to be addressed simultaneously:

---

<sup>1</sup>Similar amplification of vacuum solutions has also been observed in other low-energy theories of string theory [4].

1. *Sampling UV data (compactification data)*: zooming in on Calabi-Yau (CY) compactifications, large datasets of compactification spaces have been constructed with the most prominent example being the Kreuzer-Skarke (KS) database of 4-dimensional reflexive polytopes [7]. To obtain the associated EFTs, the relevant information about these compact dimensions has to be extracted. This includes the choice of geometry, its topological data like intersection numbers as well as suitable D-brane/O-plane setups satisfying tadpole and anomaly cancellation conditions. This UV data has been constructed on an individual basis in the past (see [8–11] for examples in Type IIB string theory), and tools have been developed to enable the study of large numbers of examples [12–14]. By now this provides access to ‘all’ models in the KS database which is the first step to systematically analyse their vacuum structures.
2. *Calculating EFT properties (quantum corrections)*: consistently identifying perturbative and non-perturbative effects associated with a given UV sample is a prerequisite for proper control over the EFT. For CY compactifications, this includes in particular the scalar potential for the moduli fields. For these setups, several interesting phenomenological implications (most prominently an additional phase of matter domination in the early Universe) have been established in concrete geometric constructions (see [6] for a recent review). On the methodological side, following over thirty years of development, we now have tools available for computing a wide range of complex structure potentials [15–18]. In addition, to control other moduli (e.g. Kähler moduli and open string moduli), further effects have to be considered such as non-perturbative effects via gaugino condensation or D-brane instantons (see [19] for a review).

Many explicit computational methods for various aspects of the EFT are currently under development. This includes for instance Pfaffian pre-factors in non-perturbative superpotentials (see [20–24] for recent advances) or 1-loop corrections to the Kähler potential in 4D  $\mathcal{N} = 1$  compactifications [25–27]. Even more importantly, the status of  $\alpha'$  corrections in  $\mathcal{N} = 1$  compactifications is of major concern for the success of Kähler moduli stabilisation procedures like KKLT [28] and, in particular, the LARGE volume scenario [29], see e.g. [30–33] for progress using F-theory dualities.

3. *EFT analysis*: for many explicit constructions, analysing the EFTs has been a bottleneck due to a plethora of moduli fields which effectively limits our access to string theory solutions. For example, typical examples in the KS database feature  $\mathcal{O}(100)$  complex structure and/or Kähler moduli fields with the Hodge numbers being bounded by  $h^{p,q} \leq 491$ . For these systems, analytic approaches are generically impossible except under certain assumptions.<sup>2</sup> In general, one faces two significant bottlenecks: on the one hand, the EFT analysis usually has to be hard-coded on an example by example level. On the other hand, the numerical optimisation appeared to be inefficient [34, 35].

---

<sup>2</sup>See [20] for SUSY AdS vacua with  $h^{1,1} \geq 51$ .

In this work, we focus on developing tools which naturally combine the data products from the first two of the above points and, in light of the third item, address the problem of efficiently sampling string vacua. Specifically, we focus on Type IIB flux vacua which are typically used as starting points for many moduli stabilisation procedures [18, 28, 29]. We construct UV input from smooth CY orientifolds where no complex structure or Kähler moduli are projected out. Furthermore, we focus on the large complex structure regime in moduli space where the EFT data is easily derived from string dualities [16, 36, 37]. Our examples are obtained from the KS database using `CYTools` [12] by taking advantage of recent advances in mirror symmetry [24] and orientifold constructions [13, 14].

Our main objective is a self-contained framework for computing EFT properties in the aforementioned compactifications. Using automatic differentiation, we connect elementary inputs such as the prepotential  $F$  for the complex structure moduli with the Kähler potential  $K$  and superpotential  $W$ . Subsequently, this allows us — again using automatic differentiation — to calculate the scalar potential and its derivatives. In our implementation we make heavy use of the just-in-time compilation and parallelisation features of JAX [38] which, as discussed later, accelerates our code in comparison to a simple python implementation by more than two orders of magnitude. We highlight that, although we provide a framework for type IIB flux compactifications, all components of our numerical implementation are modular. Little effort is required to build extensions including further aspects of low energy EFTs or exploring different string duality frames.

In our pipeline, we are able to generate efficiently the relevant equations (like the  $F$ -flatness conditions  $D_I W = 0$ ) associated with string solutions for any UV input. This defines an underlying optimisation problem which is the main use case in this work. We solve the latter using standard optimisation methods readily available via standard libraries. Arguably, solving this optimisation problem and generating solutions numerically is a challenging task by itself, but it becomes even more difficult to locate *trustworthy* and *phenomenologically relevant* solutions. Regarding the former, this means that we seek to find solutions in regimes of moduli space where calculations are under control. Equally important, we constrain our search to those values of the tadpole that can be realised in actual orientifold compactifications of Type IIB superstring theory. If we allowed unconstrained tadpoles, we would naturally find solutions almost anywhere for any number of moduli simply because of the drastically increased density of vacua in our search space.

We stress that, apart from the minimising task itself, evaluating and checking the aforementioned constraints on, say, billions of fluxes and the associated solutions to F-term equations is computationally expensive. The study of such large ensembles is necessary to address phenomenological requirements and to be able to statistically test the properties of the string landscape in a meaningful manner. Our framework naturally connects with other ongoing work to develop methods which allow to efficiently search for phenomenologically viable solutions in the string flux landscape [39–42].

The rest of this paper is organised as follows. In section 2 we give a short portrait of CY orientifold flux compactifications and introduce the type of optimisation problems to be solved in the remainder of the paper. Section 3 describes our implementation and numerical approach to finding flux vacua. We present detailed numerical experiments in section 4 and summarise our results in the conclusions in section 5.

## 2 Type IIB flux compactifications

In this section, we review Type IIB flux compactifications on CY orientifolds and the type of equations that need to be solved for flux vacua. We mainly state results to set our notation and refer to the literature for more detailed discussions (see [43–45] for reviews). These considerations are prerequisites for the subsequent implementation of our root finding and sampling methods.

### 2.1 Type IIB flux vacua

Compactifying Type IIB superstring theory on a CY orientifold leads to 4D  $\mathcal{N} = 1$  supergravity with  $h_+^{1,1}$  Kähler and  $h_-^{1,2}$  complex structure moduli  $Z^i$ . The classical Kähler potential  $K$  for the  $Z^i$  and the axio-dilaton  $\tau$  can be written as

$$K = -\log(-i\Pi^\dagger \cdot \Sigma \cdot \Pi) - \log(-i(\tau - \bar{\tau})) , \quad \Sigma = \begin{pmatrix} 0 & \mathbb{1} \\ -\mathbb{1} & 0 \end{pmatrix} . \quad (2.1)$$

Here, we defined the period vector  $\Pi$  in terms of the prepotential  $F$  as

$$\Pi = \begin{pmatrix} 2F - Z^i F_i \\ F_i \\ 1 \\ Z^i \end{pmatrix} , \quad F_i = \partial_{Z^i} F . \quad (2.2)$$

The presence of 3-form fluxes induces the  $F$ -term scalar potential

$$V_{\text{Flux}} = e^K \left( K^{\tau\bar{\tau}} D_\tau W D_{\bar{\tau}} \bar{W} + K^{i\bar{j}} D_i W D_{\bar{j}} \bar{W} \right) \quad (2.3)$$

in terms of the Gukov-Vafa-Witten (GVW) superpotential [17]

$$W = (f - \tau h)^T \cdot \Sigma \cdot \Pi(Z) , \quad (2.4)$$

and integer flux quanta

$$f = (m_J, m^I) , \quad h = (n_J, n^I) , \quad I, J = 0, \dots, h_-^{1,2}(X_3) . \quad (2.5)$$

For given choices of a 3-form flux background  $G_3 = f - \tau h$ , the potential (2.3) exhibits a non-trivial structure of minima called *flux vacua*. In the first instance it is these flux vacua which we aim at identifying. The available choices of fluxes are limited because they contribute to the cancellation condition for the  $C_4$ -tadpole,

$$\frac{1}{2} N_{\text{flux}} + N_{D3} - N_{\bar{D3}} = \frac{Q_{D3}}{2} , \quad Q_{D3} = \frac{\chi(Y_4)}{12} , \quad (2.6)$$

in terms of the D3-brane charge induced by fluxes

$$N_{\text{flux}} = \int_{X_3} H_3 \wedge F_3 = f \cdot \Sigma \cdot h = m_I n^I - m^I n_I . \quad (2.7)$$

where  $N_{D3}$  ( $N_{\overline{D3}}$ ) denotes the number of (anti-)D3-branes. The quantity  $\chi(Y_4)$  is the Euler characteristic of the  $F$ -theory fourfold [46] encoding D3-charge contributions from O-planes and D7-branes.

In this work, we focus on a class of flux vacua for which the  $F$ -term conditions  $D_{\Phi^I} W = D_I W = 0$ ,  $\Phi^I \in \{\tau, Z^i\}$ , are satisfied.<sup>3</sup> One finds that

$$D_{\tau} W = \frac{1}{\bar{\tau} - \tau} (f - \bar{\tau} h)^T \cdot \Sigma \cdot \Pi(Z) = 0, \quad (2.8)$$

$$D_i W = (f - \tau h)^T \cdot \Sigma \cdot (\partial_i \Pi(Z) + \Pi(Z) \partial_i K) = 0. \quad (2.9)$$

In fact, these two conditions are equivalent to saying that the 3-form  $G_3$  must be imaginary self-dual (ISD), i.e.,  $\star_6 G_3 = i G_3$  in terms of the Hodge star operator  $\star_3$  on  $X_3$ . We can express this condition more explicitly as [47]

$$m_J - \tau n_J = \bar{N}_{JI} (m^I - \tau n^I) \quad (2.10)$$

in terms of the gauge kinetic matrix

$$\mathcal{N}_{IJ} = \bar{F}_{IJ} + 2i \frac{\text{Im}(F_{IL}) X^L \text{Im}(F_{JK}) X^K}{X^M \text{Im}(F_{MN}) X^N}, \quad F_{IJ} = \partial_{X^I} \partial_{X^J} F. \quad (2.11)$$

For ISD fluxes,  $N_{\text{flux}} \geq 0$  is non-negative which is why sources with negative D3-brane charge like D-branes and O-planes [18] are required for tadpole cancellation (2.6).

Flux choices can lead to equivalent vacua by being identified under transformations of  $\mathcal{G} = \text{SL}(2, \mathbb{Z}) \times \Gamma$  where the former is the Type IIB S-duality group and  $\Gamma$  the modular group acting on  $\mathcal{M}_{\text{cs}}(X_3)$ . This can be avoided by mapping solutions to the fundamental domain of  $\text{SL}(2, \mathbb{Z})$ . For later convenience, let us define the VEV of the superpotential as<sup>4</sup>

$$W_0 = \sqrt{\frac{2}{\pi}} \left\langle e^{K/2} W \right\rangle, \quad (2.12)$$

which is manifestly invariant under  $\text{SL}(2, \mathbb{Z})$  transformations.

We note that for models with  $h^{1,2} > 1$ , very little is known about the solution space of string theory, see however [35, 50, 51] for examples with  $h^{1,2} \geq 4$  and [34] for  $h^{1,2} = 2$ . A key obstacle hereby is systematically solving  $F$ -term conditions (2.8), (2.9) for generic choices of fluxes below tadpole. Typically, analytic methods cease to work in regimes with  $h^{1,2} > 1$  unless for special classes of vacua like in [49, 52–54] where a subset of VEVs can be fixed analytically under certain conditions.

## 2.2 Explicit models at large complex structure

In explicit examples, we need to compute the period vector  $\Pi$  entering (2.8), (2.9). These periods can be computed by solving Picard-Fuchs equations (cf. [16, 36, 37]), employing localisation techniques (cf. [55]) or using asymptotic Hodge theory (cf. [56]).

An important class of models concerns *Large Complex Structure* (LCS) limits in complex structure moduli space  $\mathcal{M}_{\text{cs}}(X_3)$ . For such scenarios, the analytic structure for the

<sup>3</sup>The study of general vacua (e.g. non-supersymmetric solutions with  $D_I W \neq 0$ ) is left for the future.

<sup>4</sup>The convention for the normalisation is chosen based on [48, 49].

prepotential is well understood and easily computed using mirror symmetry, see [24] for recent progress. In fact, around LCS points, the coordinates  $Z^i$  of  $\mathcal{M}_{\text{cs}}(X_3)$  are identified with the Type IIA Kähler moduli in the large volume limit of Type IIA string theory compactified on the mirror dual CY manifold  $\tilde{X}_3$  [36, 57, 58]. This identification implies that the complex structure moduli  $Z^i$  take values in the Kähler cone of  $\tilde{X}_3$

$$\mathcal{K}_{\tilde{X}_3} = \{J \in H^{1,1}(\tilde{X}_3, \mathbb{R}) : \text{Vol}_J(U) > 0 \forall \text{ sub-varieties } U\} \subset H^{1,1}(\tilde{X}_3, \mathbb{R}), \quad (2.13)$$

where the sub-varieties are effective curves, effective divisors and  $\tilde{X}_3$  itself. It describes the moduli space of Kähler structures on  $\tilde{X}_3$  parametrised by a Kähler form  $J$ . By abuse of terminology, we simply refer to  $\mathcal{K}_{\tilde{X}_3}$  as the Kähler cone without mentioning the mirror side explicitly.

The prepotential computed from mirror symmetry reads [36, 57, 58]

$$F = -\frac{1}{6} \kappa_{ijk} Z^i Z^j Z^k + \frac{1}{2} a_{ij} Z^i Z^j + b_i Z^i + \frac{i}{2} \tilde{\xi} + F_{\text{inst}}(Z). \quad (2.14)$$

Here, the parameters  $\kappa_{ijk}$  are the triple intersection numbers of  $\tilde{X}_3$  which, along with the other parameters, are defined as

$$\begin{aligned} \kappa_{ijk} &= \int_{\tilde{X}_3} J_i \wedge J_j \wedge J_k, & a_{ij} &= \frac{1}{2} \int_{\tilde{X}_3} J_i \wedge J_j \wedge J_j \bmod \mathbb{Z}, \\ b_j &= \frac{1}{4!} \int_{\tilde{X}_3} c_2(\tilde{X}_3) \wedge J_j, & \tilde{\xi} &= \frac{\zeta(3) \chi(\tilde{X}_3)}{(2\pi)^3}. \end{aligned} \quad (2.15)$$

Here,  $c_2(\tilde{X}_3)$  denotes the second Chern class of the mirror manifold  $\tilde{X}_3$ . Further, the  $J_i \in H^{1,1}(\tilde{X}_3, \mathbb{Z})$  are (1,1)-forms and  $\chi(\tilde{X}_3)$  is the Euler characteristic of  $\tilde{X}_3$ .

Finally, the worldsheet instantons on the mirror dual side give rise to exponential contributions of the form [36, 58]

$$F_{\text{inst}}(Z^i) = -\frac{1}{(2\pi i)^3} \sum_{q \in \mathcal{M}(\tilde{X}_3)} n_q^{(0)} \text{Li}_3\left(e^{2\pi i q_i Z^i}\right), \quad \text{Li}_3(x) = \sum_{m=1}^{\infty} \frac{x^m}{m^3} \quad (2.16)$$

in terms of genus zero *Gopakumar-Vafa (GV) invariants*  $n_q^{(0)}$  [59, 60] of effective curves  $q$  in the *Mori cone*  $\mathcal{M}(\tilde{X}_3)$  of the mirror manifold  $\tilde{X}_3$ . It turns out to be more convenient to work with a different set of invariants obtained from a resummation of poly-logarithms. These are the so-called (genus zero) *Gromov-Witten (GW) invariants*  $N_q^{(0)}$  which are related to the GV invariants via

$$\sum_{q \in \mathcal{M}(\tilde{X}_3)} n_q^{(0)} \text{Li}_3\left(e^{2\pi i q_i Z^i}\right) = \sum_{q \in \mathcal{M}(\tilde{X}_3)} N_q^{(0)} e^{2\pi i q_i Z^i}. \quad (2.17)$$

A systematic approach to computing these invariants has been established by HKTY [16, 36]. In practice, we compute the GV and GW invariants using `CYTools` [12, 24] up to some finite degree.

The validity of the ansatz (2.14) for  $F$  is limited to the region of convergence of the LCS expansion [58]. The radius of convergence is determined by the singularity of the

associated Yukawa couplings [61, 62]. In this paper, we only check mild conditions to ensure the validity of our solutions up to a given cutoff on the GV invariants. That is, we look at solutions for which

$$\left| \frac{F_{\text{inst}}}{F_{\text{pert}}} \right| < \varepsilon \quad (2.18)$$

for some  $\varepsilon$  where  $F_{\text{inst}}$  is computed up to some finite cutoff in the instanton expansion. In practice, we typically choose  $\varepsilon = 0.1$ , though we also investigate the cutoff dependence of the instanton expansion by including higher order contributions to  $F_{\text{inst}}$  in the region of interest.

Before we continue, let us stress that going beyond the LCS regime is possible and should be part of future investigations. Here we limit ourselves to the LCS regime as computational tools from mirror symmetry are readily available in this regime [24]. An extension of these tools to other regimes is a clear direction for future developments (see [63, 64] for work along those lines).

### 2.3 Model construction and data-sets

Having presented the framework for our optimisation problem, we briefly comment on the data-sets which will be studied subsequently.

**CY and orientifold construction.** Foremost, we focus on the Kreuzer-Skarke (KS) database [7] comprised of 473,800,776 reflexive polytopes in four dimensions. Any fine, regular, star triangulation (FRST) of these polytopes leads to a CY manifold embedded as the anti-canonical hypersurface [65]. Given that a single polytope can have many FRSTs, it typically features several non-isomorphic CY manifolds, though the actual number of topologically inequivalent ones remains opaque.<sup>5</sup>

Orientifolds are constructed from  $\mathbb{Z}_2$ -involutions of Calabi-Yau manifolds. In the context of the KS database, the fixed point loci can often be obtained from simple conditions on the polytope [13, 14]. Although such involutions are inherited from the ambient toric variety covering only a subset of all involutions, they allow for the construction of a sizable number of orientifolds. We note that, in some cases, orientifold involutions lead to singularities which have to be resolved appropriately (see [68, 69] for detailed discussions). For the examples discussed in this paper, we checked that our orientifolds do not give rise to any such singularities.

For convenience, we consider orientifolds with  $h_-^{1,1} = h_+^{1,2} = 0$  for which the D7-tadpole is cancelled locally by putting D7-branes on top of the O7-planes. The D3-charge from O-planes and non-Higgsable  $\text{SO}(8)$ -stacks is given by

$$Q_{\text{D3}} = 2 + h^{1,1} + h^{1,2}. \quad (2.19)$$

---

<sup>5</sup>There exist rigorous upper bounds on the number of homotopically inequivalent CYs from polytopes in the KS database [66] which have been obtained by bounding the number of inequivalent, fine 2-face triangulations and applying Wall's theorem [67].

Throughout the main text, we constructed suitable models from the KS database using `CYTools` [12] making sure that there exists an orientifold with  $h_+^{1,2} = 0$ .<sup>6</sup> Ultimately, this requirement ensures that the orientifold intersects the LCS patch allowing us to directly apply the formulas stated in previous section.

**Flux vacua at LCS.** In this work, we are concerned with analysing large samples of vacua to investigate their distributions and attainable properties. It is clear that, due to the absence of generic analytic solutions, this requires dedicated numerical tools. On the aforementioned geometries, the tadpole (through (2.19)) and hence the possible flux vacua are bounded [2, 70]. This number is generally speaking huge and estimates obtained from continuous flux approximations suggest that the number of flux vacua with  $D_I W = 0$  and  $N_{\text{flux}} \leq Q_{D3}$  is given by [2]

$$\mathcal{N}(N_{\text{flux}} \leq Q_{D3}) = \frac{(\pi Q_{D3})^{b_3}}{b_3! \pi^{b_3/2}} \int_{\mathcal{M}_{\text{cs}}} \det(\mathcal{R} + \omega). \quad (2.20)$$

Here, we defined the 3rd Betti number  $b_3 = 2(h^{1,2} + 1)$ , the curvature 2-form  $\mathcal{R}$  and the Kähler form  $\omega$  on moduli space. We refer to [71] for an explicit evaluation of this formula for examples with few moduli.

We stress that the absence of solutions for a given choice of fluxes in our subsequent routine has to be seen in conjunction with the restriction to the LCS regime and to a single Kähler cone phase associated with a single FRST of a polytope. To give some heuristic motivation, let us assume that for each of the  $N$  moduli half of the parameter range falls within the strict LCS regime  $\text{Im}(Z^i) > 0$ . This implies that at large  $N$  a uniformly sampled point in moduli space has a probability of  $1/2^N$  to fall in the LCS regime. In reality, the angles between two hyperplanes of the cone becomes smaller as we scale up  $h^{1,2}$  making the cone even narrower than our previous naive estimate [72].

On top of that, even if points can be efficiently sampled from  $\mathcal{K}_{\tilde{X}_3}$ , the actual question to ask is: given a choice of fluxes, what are the chances of finding a minimum within the LCS patch traced out by  $\mathcal{K}_{\tilde{X}_3}$ ? In our case, we search for flux vacua in a single toric phase defined by a single FRST of a polytope. Generally, this geometric phase only covers a small subspace of moduli space where the existence of solutions to (2.8), (2.9) for arbitrary flux choices with  $N_{\text{flux}} \leq Q_{D3}$  cannot be guaranteed.<sup>7</sup> As we discuss later in section 4.3, this seriously affects our ability to sample vacua at large values of  $h^{1,2}$ .

### 3 Algorithmic approach for string vacua

Below we outline the scope and functionalities of our current numerical approach. Our code is modular such that each component can be developed further on an individual basis and allows to be integrated into the pipeline to extract physical properties. In short, the rationale is as follows:

---

<sup>6</sup>We thank Jakob Moritz for explaining us the necessary conditions on polytopes allowing for orientifolds with  $h_+^{1,2} = 0$ .

<sup>7</sup>We refer to [73] for related discussions in the context of 5D black hole solutions in M-theory.

- (i) *Model construction*: we start with a high level input which can either be a custom example or directly interface with existing databases such as the KS database or the CICY database [74, 75].
- (ii) *EFT module*: we calculate relevant quantities in the 4D supergravity EFT starting from a prepotential. Here, this includes the flux superpotential (2.4) and the Kähler potential (2.1) for complex structure moduli. Subsequently, we use the known analytical framework to calculate quantities like the  $F$ -term scalar potential (2.3). This pipeline is implemented such that we can ensure scalability in the presence of many moduli. To do this, we utilise JAX and its just-in-time (`jit`) and vectorisation (`vmap`) features. As detailed below, compared to alternative implementations in `python` or `Mathematica`, simply evaluating relevant quantities with our methods is faster by several orders of magnitude, see figure 1.
- (iii) *Optimisation module*: we formulate the optimisation conditions for identifying minima of moduli potentials and pass them to optimisation algorithms. A priori there is no optimisation algorithm which is singled out. At this stage, we restrict ourselves to the use of `scipy.optimize.root` which we find to be rather efficient for our purposes.
- (iv) *Sampling module*: of particular importance for the efficiency of our optimisation algorithm is the choice of initial guesses for the moduli VEVs, i.e., from where the algorithm starts searching for minima. We present novel methods to choose these points based on the choice of flux quanta using the ISD condition (2.10).
- (v) *Filter module*: the minima are identified using a certain numerical tolerance of the respective conditions. To warrant that we are dealing with actual minima, we perform additional checks on these candidates as outlined below. Among others, this involves checking the positivity of the Hessian of the scalar potential.

Below, we describe in more detail each of the individual modules (ii)-(v) building our framework for the search for string vacua. (The model construction (i) is largely outsourced using tools readily available in `CYTools`.) In addition, we provide some pseudo-code snippets in appendix A. Again we stress that each of these components could be modified without interfering with the other parts of the code. Given our implementation, our current bottleneck when extrapolating to large  $h^{1,2}$  is the sampling and root finding which we comment on in due course.

### 3.1 EFT module — moduli potentials in JAX

For our implementation, we opted for JAX [38] which at its heart employs automatic differentiation to compute gradients of functions. It recognises primitive operations inside functions and applies standard rules for differentiation in such a way to numerically evaluate derivatives rather efficiently. In particular, it avoids finite differences to compute approximate derivatives in numerical differentiation, and it bypasses inefficient expressions in symbolic differentiation which would appear for instance when using `sympy`.

Overall, this makes the implementation differentiable, i.e., derivatives of quantities with respect to intermediate or input variables can be taken and utilised at machine precision. This is ideal for our quest for finding string vacua in 4D supergravity where only a handful of functions are prerequisites, in our case the prepotential (2.14), in order to compute quantities like period vectors, scalar potentials or the Hessian by taking suitable derivatives. In particular, this implies that our code only needs the absolute minimum of functions to be hard-coded, thereby making our implementation extremely versatile. Among others, it is readily applicable to any moduli space limit away from LCS such as conifold regions and, with minor adjustments, can be easily extended to include Kähler moduli.

Apart from the usefulness of auto-differentiation, there are two powerful tools to speed up the implementations. The first is `jit`-compilation (standing for just-in-time compilation) which, in short, transforms `python`-functions into language representations via so-called *tracer objects* that record all the operations being performed. This reconstruction improves the execution by sending full sequences of operations to tensorflow’s XLA (Accelerated Linear Algebra) compiler [76], while in standard `python`-code each operation is sent to the compiler one at a time.<sup>8</sup> For our purposes, `jit` ensures that there is only a modest scaling with respect to the number of moduli fields through essentially C++ speed of the code. Additionally, the usage of `JAX` makes most modules readily usable on the GPU which can further improve the efficiency.

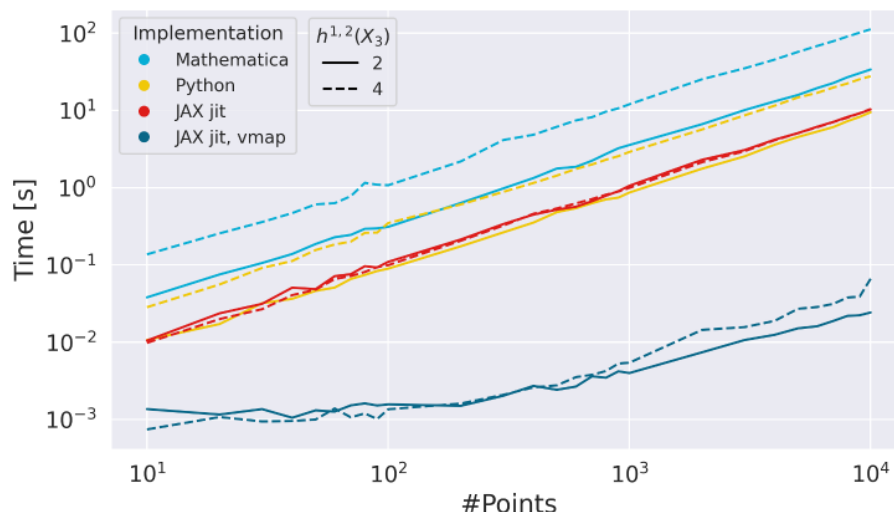
Secondly, while vectorisation can always be implemented manually (by rearranging indices and axes), this can become increasingly tedious when the range or the number of indices changes dynamically. This is where the automatic vectorisation feature `vmap` of `JAX` is particularly useful. For example, we may want to evaluate the  $F$ -term conditions for a single flux at many points or, instead, for several fluxes, but for each flux at a different collection of points. Both cases can be addressed by calling `vmap` on the various inputs (see the [JAX documentation](#) for examples). Clearly, the upshot of vectorising expressions in our implementation are enormous gains in efficiency as we now illustrate.

Let us quantify the improvements of our implementation compared to popular alternatives, namely `Mathematica` (version 13.4) and `python` (version 3.9). We compare the time required to evaluate the  $F$ -term conditions  $D_I W$  for all fields in different implementations in figure 1. We present two models with  $h^{1,2} = 2, 4$  complex structure moduli using in both cases GV invariants up to degree 4.<sup>9</sup> In `Mathematica` and `python`, we used a hard-coded version of the  $F$ -term conditions, while our `jit`- and `vmap`-compilation starts from the prepotential (2.14) and constructs the  $F$ -terms via automatic differentiation. We use `JAX` version 0.4.4 in our benchmark comparison.

In both the `Mathematica` and `python` implementations, one observes a clear scaling with  $h^{1,2}$  coming from the increasing number of primitive operations required when computing  $D_I W$ . In contrast, the implementations with `jit` show virtually no difference between

<sup>8</sup>We note however `jax.jit` cannot be applied to any arbitrary function, see the official [JAX documentation](#). For instance, whenever specific values are tested as conditions (e.g., via `if`-statements), the `jit`-compilation is not directly applicable but generally workarounds can be identified.

<sup>9</sup>The two models in question are  $\mathbb{P}[1, 1, 1, 6, 9]$  for  $h^{1,2} = 2$  discussed in section 4.1 and the first example in [35] for  $h^{1,2} = 4$ .



**Figure 1.** *Efficiency comparison of different implementations.* Time required to evaluate the  $F$ -term conditions for two models, described in the text, with  $h^{1,2} = 2, 4$  at a given number of points using different implementations. Each curve is obtained from averaging over 10 runs performed on the same laptop using a single core. Further improvements are easily obtained when parallelising over multiple cores or using our code on the GPU.

the two models. As promised above, the most dramatic speed up is however observed by making use of `vmap` which improves the evaluation time by roughly two orders of magnitude. A comparison with other implementations like a direct C++ implementation is beyond the scope of this paper.

Before we continue, we note that, while we provide implementations for general supergravity equations, it can occasionally be more efficient to use simplified expressions. For example, say we solved the  $F$ -flatness conditions for a given choice of fluxes. Then it can be useful to employ identities for the  $F$ -term scalar potential (2.3) and related quantities like the Hessian, see e.g. [77].

### 3.2 Optimisation module — numerical search for extremal points

In our quest for string vacua, we would like to numerically solve  $D_I W = 0$  for given inputs of UV data. Since most optimisation algorithms work with real-valued variables we re-formulate our complex-valued optimisation conditions (cf. eqs. (2.8), (2.9)) in terms of their respective real and imaginary parts. We end up with a one-dimensional array of size  $2(h^{1,2} + 1)$  for which we find the roots in terms of the real and imaginary parts of the moduli fields.<sup>10</sup>

For our analyses, finding zeros of  $D_I W$  is efficiently achieved using the root finding methods of `scipy` [78], especially when compared with homotopy solvers previously employed in [34, 35]. From the methods implemented within `scipy.optimize.root`, we determine via direct comparisons that `method='hybr'` associated with a modified version

<sup>10</sup>We note that empirically, we find that using the analytic solution for the dilaton obtained from solving (2.8) does not simplify the optimisation substantially.

of Powell’s method [79] works most reliably and efficiently. As input, we provide a choice of integer fluxes together with initial guesses for the roots. We comment on the sampling of both in the subsequent section. In practice, we run the optimisation module on several CPUs in parallel to speed up the computation. As numerical tolerance  $\epsilon_{\text{tol}}$  for the root finding, we use  $\epsilon_{\text{tol}} = 10^{-10}$  across all models which was chosen as a robust choice when comparing with known special solutions for the torus [80] and  $\mathbb{P}[1, 1, 1, 6, 9]$  [49, 81].

Beyond `scipy.optimize.root` there are several optimisation methods which could be used. At this stage, however, involving only a small number of checks, we did not find a more efficient method when comparing with e.g. gradient descent approaches. Clearly, choosing eqs. (2.8), (2.9) as optimisation targets is only one possibility and our framework can be easily applied to the gradient of the scalar potential instead. This is possible due to the efficient automatic differentiation capabilities of our implementation. In the long term, it would be beneficial to apply even more efficient optimisers which can be parallelised easily within the JAX framework and make use of the GPU. These modifications will be the target of the next round of improvements of our framework.

### 3.3 Sampling module — sampling of fluxes and initial guesses

In this section, we detail the strategies to efficiently sample flux choices and starting conditions for our root finding algorithm. As we will show subsequently, this vastly outperforms previous access to solutions. Hereby, it is absolutely crucial to employ `vmap` to guarantee speedups of several orders of magnitude similar to figure 1.

Obviously, there remain biases associated to our selection procedure. Although we believe that one should account for these biases using existing statistical techniques, this analysis goes beyond the current scope of the paper.<sup>11</sup> Here, our main focus is to establish novel methods to numerically access these vacua. The various sampling techniques to be introduced below will be compared in the subsequent section, see in particular figure 2.

**Initial guesses.** Initially, we need to specify points in moduli space from which to start the root finding. To this end, we define a region, e.g. sphere, box, or the Kähler cone  $\mathcal{K}_{\tilde{X}_3}$ , in which points are being uniformly sampled. We treat the size of these regions as a hyperparameter. While the notion of the Kähler cone is mainly useful for LCS limits, conifold regions can be described by spheres centered around the origin.

For our considerations, it is useful to work inside subcones of  $\mathcal{K}_{\tilde{X}_3}$  defined in (2.13), namely so-called *stretched* Kähler cones defined for some  $c \in \mathbb{R}_+$  via [72, 82]

$$\mathcal{K}_{\tilde{X}_3}[c] = \{J \in H^{1,1}(\tilde{X}_3, \mathbb{R}) : \text{Vol}_J(U) \geq c \ \forall \text{ sub-varieties } U\}. \quad (3.1)$$

The tip of  $\mathcal{K}_{\tilde{X}_3}[c]$  is given by the shortest (in the Euclidean metric) vector  $v_{\text{tip}}$  in the Kähler cone  $\mathcal{K}_{\tilde{X}_3}$  for which the  $\mathbb{Z}_+$  generators of the Mori cone  $\mathcal{M}(\tilde{X}_3)$  (as the dual of its closure) have volume at least  $c$ . It is convenient to work with  $\mathcal{K}_{\tilde{X}_3}[c]$  for some  $c > 0$

---

<sup>11</sup>An example of such algorithmic biases was discussed recently in [42].

because the convergence of  $F_{\text{inst}}$  is more easily guaranteed.<sup>12</sup> In addition, by picking initial guesses far inside  $\mathcal{K}_{\tilde{X}_3}[c]$ , we are more likely to find roots inside the domain of validity of the LCS expansion, thereby improving the stability and success rate of our root finding methods.

Lastly, we implemented a variant of the cone sampling procedure which is useful in models where the generators of  $\mathcal{K}_{\tilde{X}_3}$  are not known explicitly. In such cases, one can select initial guesses along the one-dimensional subspace  $\{c \cdot v_{\text{tip}} : c \in \mathbb{R}_+\}$  (with  $v_{\text{tip}}$  here and subsequently denoting the tip of  $\mathcal{K}_{\tilde{X}_3}[c=1]$ ) corresponding to the tips of the stretched cones for varying  $c$ .

**Flux choices.** We have to specify flux choices where we distinguish the following two approaches:

1. **We sample fluxes independently from the starting point:** we do this randomly by sampling from the uniform distribution such that  $N_{\text{flux}}$  is below or at a fixed tadpole  $Q_{D3}$  as set by eq. (2.19). Similarly, we sample starting points independently from our flux choices using one of the aforementioned techniques.
2. **ISD biased sampling:** we introduce a new sampling technique which ensures that both the flux is below the tadpole and the starting point is close to the point satisfying the ISD condition in eq. (2.10). The idea is to sample only half of the fluxes together with points  $(Z_0^i, \tau_0)$  inside the Kähler cone. Then, the ISD constraint (2.10) can be solved for the remaining half of the fluxes. We implemented two variants, namely

$$\text{ISD}_+ : \quad \tilde{m}_J - \tau_0 \tilde{n}_J \equiv (\overline{\mathcal{N}}_0)_{JI} (m^I - \tau_0 n^I), \quad m^I, n^I \in \mathbb{Z}, \quad (3.2)$$

$$\text{ISD}_- : \quad \tilde{m}^I - \tau_0 \tilde{n}^I \equiv (\overline{\mathcal{N}}_0)^{IJ} (m_J - \tau_0 n_J), \quad m_J, n_J \in \mathbb{Z}. \quad (3.3)$$

Here,  $\mathcal{N}_0$  denotes the gauge kinetic matrix (2.11) evaluated at  $Z_0^i$ . We sample the fluxes on the right hand sides of (3.2) and (3.3), while solving for the fluxes on the left. We note that this method modifies a previous algorithm [83] for generating a pair of flux and starting point values by sampling points from the entire Kähler cone that achieve  $N_{\text{flux}} \leq Q_{D3}$ .<sup>13</sup> In general, the fluxes  $\tilde{m}, \tilde{n}$  (dropping indices for convenience) obtained in this way are continuous. To get a meaningful flux vacuum consistent with flux quantisation, we have to round the fluxes  $\tilde{m}, \tilde{n} \rightarrow m, n \in \mathbb{Z}$ . In effect, this implies that we no longer solve (3.2), (3.3) exactly at our initial point in moduli space, but pay the price of shifting the moduli VEVs  $\langle Z^i \rangle, \langle \tau \rangle$  slightly away from  $Z_0^i, \tau_0$ , i.e.,

$$\langle Z^i \rangle = Z_0^i + \delta Z^i, \quad \langle \tau \rangle = \tau_0 + \delta \tau. \quad (3.4)$$

<sup>12</sup>To be more precise, the convergence of  $F_{\text{inst}}$  demands that we stay far enough away from walls of  $\mathcal{K}_{\tilde{X}_3}$  where some of the (mirror) effective curve volumes become zero and the sum in (2.16) potentially diverges. Clearly, this depends on the structure of (non-zero) GV invariants. For instance, in particular examples, we can analytically continue the LCS periods to regions near conifold singularities by shrinking GV-nilpotent curves [63, 64].

<sup>13</sup>Similar strategies have previously been employed in e.g. [8].

The sizes of the induced shifts  $\{\delta Z^i, \delta\tau\}$  from rounding the fluxes depend crucially on the chosen method  $\text{ISD}_\pm$ , see figure 2. More importantly, the two sampling procedures give rise to different characteristics of the resulting distribution of ISD solutions as we will show in section 4.1.

### 3.4 Filter module — identification of trustworthy string vacua

After computing the stationary points of  $V_F$  by solving  $D_I W = 0$  or  $\partial_I V_F = 0$ , we feed them into a pipeline to extract trustworthy minima.<sup>14</sup> Currently, we implemented the following checks on our solutions:

- We ensure *positive string couplings and inequivalence to other solutions under  $SL(2, \mathbb{Z})$  transformations*. This is simply because we are interested only in physically relevant solutions with  $g_s > 0$ , while avoiding overcounting solutions due to  $SL(2, \mathbb{Z})$ . Where necessary, we map  $\tau$  to the fundamental domain of  $SL(2, \mathbb{Z})$  by applying translations and S-duality transformations.
- We verify that our solutions meet the *Kähler cone conditions*. Depending on the available information associated with the background geometry, we use different implementations to verify that the Kähler cone conditions are satisfied. If hyperplanes or generators of the cone are available, then we can easily check whether the given VEVs lie inside the Kähler cone. Alternatively, we check that the Kähler metric is positive definite which is however not a sufficient condition.
- We check that the *Hessian of the scalar potential is positive semi-definite* by computing its Cholesky decomposition as we find this method to be more efficient than computing eigenvalues directly. This guarantees that the obtained stationary points are actual minima of the scalar potential. In addition, even if non-negativity of the Hessian can be guaranteed, ensuring the *absence of flat directions* requires special attention. This is because it can be numerically hard to distinguish minima from flat direction, especially when large hierarchies of scales are involved. For vacua with  $D_I W = 0$ , the number of massless fluctuations around the minimum, i.e., the dimension of the remaining moduli space is given by the rank of the matrix<sup>15</sup> (see e.g. [77])

$$M|_{D_I W=0} = \begin{pmatrix} D_I D_J W & K_{I\bar{J}} W \\ K_{\bar{I}J} \bar{W} & D_{\bar{I}} D_{\bar{J}} \bar{W} \end{pmatrix}. \quad (3.5)$$

For vacua with  $D_I W \neq 0$ , we have to consider the number of first order obstructions to  $\partial_I V_F = 0$  which is simply the rank of the Hessian. In both cases, the rank is computed using the tolerance  $\epsilon_{\text{tol}}$  used in the optimisation module.

<sup>14</sup>It is certainly interesting to analyse the set of all stationary points and compare against results from random matrix theory such as [51, 84, 85]. We will come back to these questions in the future.

<sup>15</sup>The diagonal entries of this matrix (evaluated at a minimum) are proportional to the mass matrix of chiral fermions.

| $d_1 \backslash d_2$ | 0   | 1              | 2               | 3               | 4                  |
|----------------------|-----|----------------|-----------------|-----------------|--------------------|
| 0                    | 0   | 3              | -6              | 27              | -192               |
| 1                    | 540 | -1,080         | 2,700           | -17,280         | 154,440            |
| 2                    | 540 | 143,370        | -574,560        | 5,051,970       | -57,879,900        |
| 3                    | 540 | 204,071,184    | 74,810,520      | -913,383,000    | 13,593,850,920     |
| 4                    | 540 | 21,772,947,555 | -49,933,059,660 | 224,108,858,700 | -2,953,943,334,360 |

**Table 1.** The leading order GV invariants for  $\mathbb{P}[1, 1, 1, 6, 9]$ .

- *Validity of the LCS approximation:* we check eq. (2.18) for  $\varepsilon = 0.1$ . This gives a rough estimate on the radius of convergence for degrees available in each model.<sup>16</sup>

A solution passing these criteria we refer to as a flux solution or vacuum. All of the above checks are fully `vmap` compatible (recall section 3.1).

## 4 Numerical experiments

Having described our algorithm, we now showcase some initial capabilities and comment on physical implications of the respective results. To begin — using the well-studied example of a degree 18 hypersurface in the weighted projective space  $\mathbb{P}[1, 1, 1, 6, 9]$  — we discuss the differences in our sampling procedures. We then generate large samples of vacua for four and five moduli models which were introduced in [35] featuring at least one million flux vacua, vastly outperforming the handful of previously obtained vacua. Interestingly, we identify that the respective  $|W_0|$ -distributions can be well approximated by the same probability distribution. Finally we analyse the scaling behaviour with  $h^{1,2}$  by studying examples with up to 25 moduli; identifying suitable success rates and timing behaviour of our approach. Our data can be accessed on <https://github.com/ml4physics>.

### 4.1 Sampling procedures in practice — $h^{1,2} = 2$

To characterise the physics associated with the distribution of flux vacua, we need to understand the biases arising from the respective sampling techniques.<sup>17</sup> To start the discussion, we present the difference of the vacua distribution in our sampling techniques previously introduced in section 3.3. To allow for simpler visualisation, we study the degree 18 hypersurface  $X_3$  in weighted projective space  $\mathbb{P}[1, 1, 1, 6, 9]$  [61] with Hodge numbers  $(h^{1,1}, h^{1,2}) = (2, 272)$ .

In detail, we focus on a particular locus in moduli space where the CY is invariant under  $G = \mathbb{Z}_6 \times \mathbb{Z}_{18}$  [86]. By restricting to fluxes invariant under this symmetry, we only need to

<sup>16</sup>Arguably, to fully trust a solution, one would need to compute the radius of convergence to high precision. We intend to discuss this problem in a separate work.

<sup>17</sup>Note that this applies to both numerically and analytically constructed flux vacua.

solve the  $F$ -term conditions along the invariant subspace. The corresponding periods can be obtained from the mirror CY giving rise to an effective prepotential depending only on two moduli. The associated topological data is

$$\kappa_{111} = 9, \quad \kappa_{112} = 3, \quad \kappa_{122} = 1, \quad a_{ij} = \frac{1}{2} \begin{pmatrix} 9 & 3 \\ 3 & 0 \end{pmatrix}, \quad b_i = \frac{1}{4} \begin{pmatrix} 17 \\ 6 \end{pmatrix}. \quad (4.1)$$

We also computed GV invariants up to degree 100 using methods described in [24] and collected the leading order GV invariants (see table 1 as a reference). We consider an orientifold with  $h_+^{1,2} = 0$  constructed explicitly in [87] for which the D3-tadpole is given by  $Q_{D3} = 276$ .

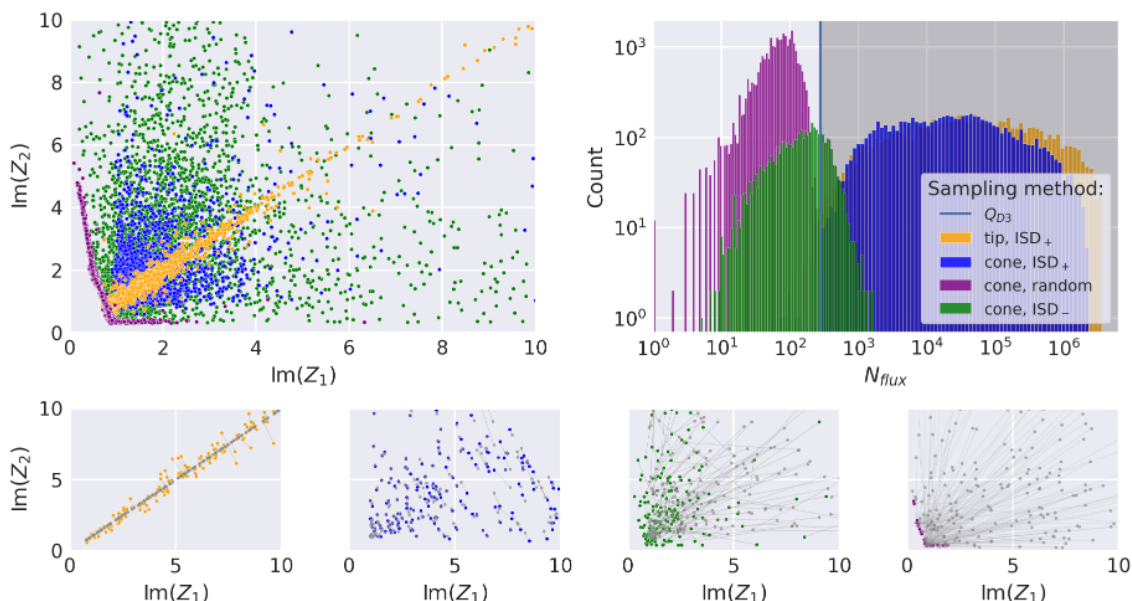
The Kähler cone  $\mathcal{K}_{\tilde{X}_3}$  is simply the positive quadrant generated by the vectors  $\{(1,0), (0,1)\}$ . One thus easily samples points inside the cone by taking positive linear combinations of these generators. The tip of the stretched cone  $\mathcal{K}_{\tilde{X}_3}[c=1]$  is just given by  $v_{\text{tip}} = (1,1)$ , recall section 3.3.

**Benchmarking our performance.** Initially, we compare our implementation with existing searches using the Paramotopy method [88] which has been employed for our model in [34] to search for flux vacua with  $N_{\text{flux}} = 34$ . While the scan of [34] was performed only including classical terms, we are able to conduct two separate searches: the first uses the purely classical prepotential, whereas the second includes instanton terms up to degree 10. We note that in our framework there is no significant change in performance for either of the two scans despite the fact that  $F_{\text{inst}}$  includes 65 additional terms with non-vanishing GVs.

Our calculations have been performed on the LMU cluster on a single node with 4 cores with 5GB of memory and 50,000 input fluxes. For the two runs, we found 34,542 (with instantons) and 33,019 (classical) vacua with  $\varepsilon = 0.1$  in (2.18) in about 45 minutes respectively. From those vacua, using sampling method  $\text{ISD}_+$ , 33,047 and 31,042 survive when choosing instead  $\varepsilon = 0.01$  in (2.18) implying that the LCS expansion is well under control in our framework. As a comparison, the equivalent scan in [34] was performed with 100 nodes each with 32 cores and required a total calculation time of around 75,000 hours to find around 24,882 acceptable solutions.

**Qualitative comparison of sampling approaches.** Turning to the differences of our sampling methods, we show the distribution of moduli VEVs for the different sampling methods in figure 2. We note that, throughout this section, we do not impose any cut on the flux contribution  $N_{\text{flux}}$  to the D3-tadpole. By sampling points only along tips of the stretched Kähler cone for different choices of  $c$  (recall (3.1)), the resulting sample of flux vacua remains largely 1-dimensional (orange). For cone sampling, one covers a much broader range inside the Kähler cone for both  $\text{ISD}_\pm$ . Interestingly, randomly sampling fluxes results in moduli VEVs clustered at the boundary of the Kähler cone. Being close to the boundary suggests that these solutions may not survive in the presence of higher order instanton corrections as the region with  $|F_{\text{inst}}/F| < \varepsilon$  shifts further inside the Kähler cone when cutting the instanton prepotential at higher degree.

Looking at the right in figure 2, we find that the distribution of flux tadpole contributions  $N_{\text{flux}}$  are vastly different among the sampling methods. We clearly see that the  $\text{ISD}_+$



**Figure 2.** Comparison between the different sampling procedures. Top left: distribution of flux vacua in the plane of moduli VEVs, utilising units as outlined in the main text. Top right: distribution of  $N_{\text{flux}}$  with the vertical line indicating the tadpole bound  $Q_{D3} = 276$  and the dark shaded region highlighting solutions with  $N_{\text{flux}} > Q_{D3}$ . Bottom row: distribution of initial guesses (gray) compared to the actual solutions to  $D_I W = 0$ .

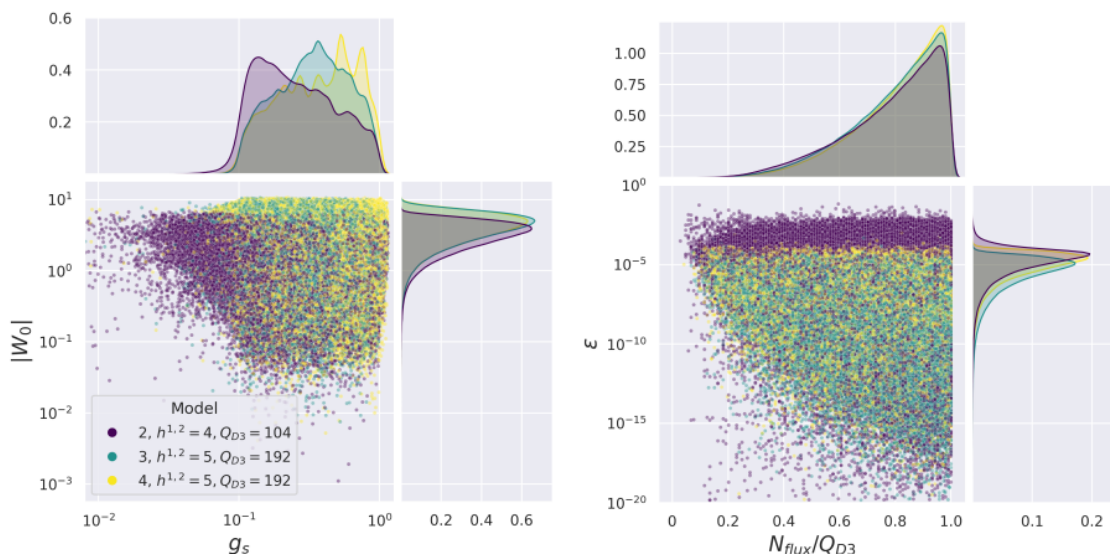
method leads to rather large tadpoles, even for models with small  $h^{1,2}$ .<sup>18</sup> In contrast, the  $\text{ISD}_-$  approach generates samples with significantly smaller  $N_{\text{flux}}$  values.

The bottom row of figure 2 shows a subset of the moduli VEVs at the minimum and the associated initial guesses. The relative distance between them depends significantly on the sampling method. These noticeable dissimilarities can lead to algorithmic biases (see [42]) as the identification of minima is subject to input parameters of the optimisation module like the maximal number of steps. For random flux sampling, we find the largest displacements where, despite sampling points far inside the cone, the moduli VEVs end up close to the boundary. In contrast, the induced shifts  $\delta Z^i$  (recall eq. (3.4)) from rounding fluxes in  $\text{ISD}_-$  sampling are appreciably smaller, though they are larger for  $\text{ISD}_-$  than for  $\text{ISD}_+$ . This is because, when using the  $\text{ISD}_-$  method, the inverse matrix  $\bar{\mathcal{N}}^{IJ}$  appears in (3.3). In effect, this amplifies the error in (3.3) when rounding the continuous fluxes on the left hand side. Hence, the actual solutions to the  $F$ -flatness conditions are on average further away from the initial guess than in  $\text{ISD}_+$ , thereby also affecting the success rate.

To conclude, we clearly see that the different biases due to the sampling method have indeed an appreciable effect. These biases have to be accounted for to deduce constraints on the space of flux vacua and to extract probabilistic statements about the string landscape.<sup>19</sup> At this stage we postpone a quantification of these effects to future work as we believe that these techniques are best applied in the context of a direct physics question.

<sup>18</sup>This is reminiscent of the observations made in [83] for a model with rather large  $h^{1,2} = 50$ .

<sup>19</sup>We note that dealing with such biases in practice is ubiquitous in observational cosmology, see for instance [89] for examples in the context of large scale structure cosmology constraints.



**Figure 3.** Samples of flux vacua with four and five complex structure moduli of [35]. *Left:* distribution of  $W_0$  and  $g_s$ , showing universal behaviour in the distribution of  $W_0$ . *Right:* distribution of  $\varepsilon$  and  $N_{\text{flux}}$ , demonstrating very good control over the instanton expansion.

In particular, let us highlight that already in this simple example the distribution of  $N_{\text{flux}}$  depends critically on the chosen sampling and solving methods. As is obvious from figure 2, large values of  $N_{\text{flux}} \gtrsim \mathcal{O}(10^3)$  are naturally preferred for  $\text{ISD}_+$  sampling. This should however not be understood as evidence for the tadpole conjecture. It is merely a sign that, due to the high dilution of string vacua with low  $N_{\text{flux}} \leq Q_{D3}$  compared to swampland solutions with high  $N_{\text{flux}} > Q_{D3}$ , randomly sampling points in  $\mathcal{K}_{\tilde{X}_3}$  generically leads to large  $N_{\text{flux}}$ . Obviously, this problem becomes even more severe at higher  $h^{1,2}$ , see also section 4.3. Hence, it becomes necessary to optimise the procedure for the sampling of points in the Kähler cone.

## 4.2 Generating samples of flux vacua — $h^{1,2} = 4, 5$

Capabilities to generate vast sets of generic flux vacua in regimes with  $h^{1,2} > 2$  has so far been rather limited. Here, we would like to demonstrate the ease with which such datasets can be efficiently produced within our framework even for more elaborate geometries. We therefore revisit models with four and five complex structure moduli presented in [35].

Specifically, we study three CY hypersurfaces presented in the appendix<sup>20</sup> of [35], though we construct orientifolds from different  $\mathbb{Z}_2$ -involutions.<sup>21</sup> The latter are obtained from the orientifold database of [90] with the D3-tadpole values  $Q_{D3} = 104$  and  $Q_{D3} = 192$  for  $(h^{1,1}, h^{1,2}) = (4, 98)$  (example 2) and  $(h^{1,1}, h^{1,2}) = (5, 185)$  (examples 3, 4) respectively.

<sup>20</sup>One easily verifies from the list of orientifolds in [90] that the model discussed in the main text of [35] does not allow for an orientifold with  $h_+^{1,2} = 0$ .

<sup>21</sup>The orientifold examples of [35] were specifically constructed to satisfy certain model building criteria presented in [11] for D-branes at singularities with  $h_-^{1,1} \neq 0$ .

As in the previous example, we study these models near a *Greene-Plesser point* [91] where the CY is manifestly invariant under a discrete subgroup  $G$ . Turning on fluxes only along the invariant 3-cycles allows to fix the non-invariant moduli fields at their fixed point under  $G$ , thereby solving the associated  $F$ -flatness conditions. The remaining periods can directly be computed on the mirror dual CY  $\tilde{X}_3$  and depend only on the invariant moduli of which we have four (example 2) or five (examples 3,4) in our models.

For these example, we now collect flux vacua by using the ISD<sub>+</sub> sampling method for the fluxes and initial points. To this end, we randomly generate fluxes in the range  $[-5, 5]$  and points inside the Kähler cone up to Euclidean distance 10 from the origin using the cone generators. For each model, we collected approximately  $10^6$  vacua taking into account instanton corrections up to degree 10 which took 10 hours of run time on a machine with 4 CPUs and 10GB memory.

We show the distribution of  $|W_0|$  and  $g_s$  on the left of figure 3. Apart from a slight shift of the peak, the former seems to be almost identical across the three models. This is a hint at universality of the  $|W_0|$  distribution across different models within our current framework. In contrast, the distribution of  $g_s$  exhibits non-trivial structures which are distinct for each model. Clearly, at this point, we are unable to infer that these are characteristic features of the respective models given the expected biases in our data, recall section 3.3. Nonetheless, it is interesting to point out that, while the distribution for example 2 peaks at small  $g_s$ , we observe two distinctive peaks at large  $g_s$  for example 4.

The distribution of  $\varepsilon$  and  $N_{\text{flux}}/Q_{D3}$  is displayed on the right of figure 3. While in the analysis of [35] most vacua were unstable against the inclusion of higher order instanton corrections, we find overall an excellent control over the instanton expansion as exhibited by small  $\varepsilon$ . This can be expected given that our sampling methods allow us to stabilise moduli deep inside the Kähler cone where instanton terms are highly suppressed. In contrast, as we have seen previously in figure 2, random sampling fluxes as in [35] tends to drive the  $F$ -term solutions closer to the boundary, thereby loosing control over the instanton expansion. This makes evident the fact that our new sampling techniques are superior to previously employed techniques.

The rescaled distribution of  $N_{\text{flux}}/Q_{D3}$  is largely model independent. At first sight, this is surprising given the expected scaling (2.20) of the number of vacua with  $h^{1,2}$  and the tadpole. We believe that this is due to the chosen sampling method, though this observation deserves further scrutiny.

### 4.3 Scaling behaviour for large numbers of moduli — $h^{1,2} \leq 25$

Having seen the capabilities to generate reliable samples, we next comment on the capabilities of this framework to analyse geometries with even larger number of moduli. To this end, we construct examples with  $h^{1,2} \in \{5, 10, 15, 20, 25\}$  moduli using **CYTools**, see table 2 and ancillary files for the details regarding these models. Specifically, we selected trilayer polytopes [14] in KS with the largest  $h^{1,1}$  for given  $h^{1,2}$  and constructed smooth  $O3/O7$  orientifolds with  $h_-^{1,1} = h_+^{1,2} = 0$ . Even though we collected these models by requiring, in addition to the existence of suitable orientifolds, large  $Q_{D3}$ , we note that similar analysis could be done with much smaller  $Q_{D3}$  at the prize of a higher run time.

| $h^{1,1}$ | $h^{1,2}$ | $Q_{D3}$ | success rate | #vacua    | $\min(N_{\text{flux}})$ |
|-----------|-----------|----------|--------------|-----------|-------------------------|
| 213       | 5         | 220      | 50%          | 1,370,842 | 5                       |
| 244       | 10        | 256      | 16%          | 498,545   | 36                      |
| 399       | 15        | 416      | 7%           | 168,291   | 116                     |
| 350       | 20        | 372      | < 1%         | 36        | 180                     |
| 245       | 25        | 272      | < 1%         | 1         | 270                     |

**Table 2.** Hodge numbers and tadpole values for the selected models.

Given the model data, we look for flux vacua with  $N_{\text{flux}} \leq Q_{D3}$ . We sample vacua by using the  $\text{ISD}_+$  method running on machines with 8 CPU cores and 12GB RAM. For each model, we include instanton contributions up to degree 2.

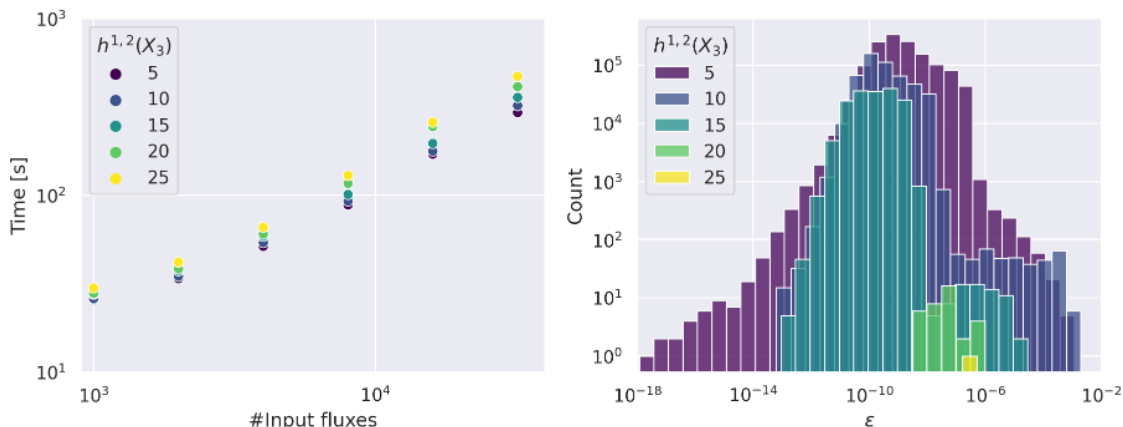
The average success rates of finding vacua from some input fluxes is listed in table 2 where we notice a significant drop at  $h^{1,2} \geq 20$ . We believe that this is because of several constraining factors conspiring to drastically decrease the success rate at large  $h^{1,2}$ . Firstly, high dimensionality of flux and moduli spaces from which we sample typically means slower evaluation time. In addition, it becomes even harder to perform numerical optimisation, i.e., to find solutions to  $D_I W = 0$ . Lastly, a single phase of the Kähler cone becomes narrower [72], while the number of phases of the extended Kähler cone increases exponentially. That is, we see less and less of the complete moduli space. This becomes e.g. relevant in ISD sampling where the induced shifts in (3.4) may easily lead to moduli VEVs  $\langle Z^i \rangle, \langle \tau \rangle$  outside the original Kähler cone. All in all, the chances get rather slim to locate minima at large  $h^{1,2}$  below a certain threshold of  $Q_{D3}$ . Clearly, despite significant progress, we point out that our methods are simply not refined enough yet to access these regimes.

In this context, we stress that sampling below a realistic value of  $Q_{D3}$  is many orders of magnitude harder than sampling with arbitrary  $N_{\text{flux}}$  by virtue of (2.20). We tested our algorithm on examples with  $h^{1,2} > 100$  and easily obtained solutions with  $N_{\text{flux}} \gg Q_{D3}$ . This is just telling us that sampling from an effectively continuous distribution for swampland solutions is much easier than probing a discrete distribution of landscape solutions.

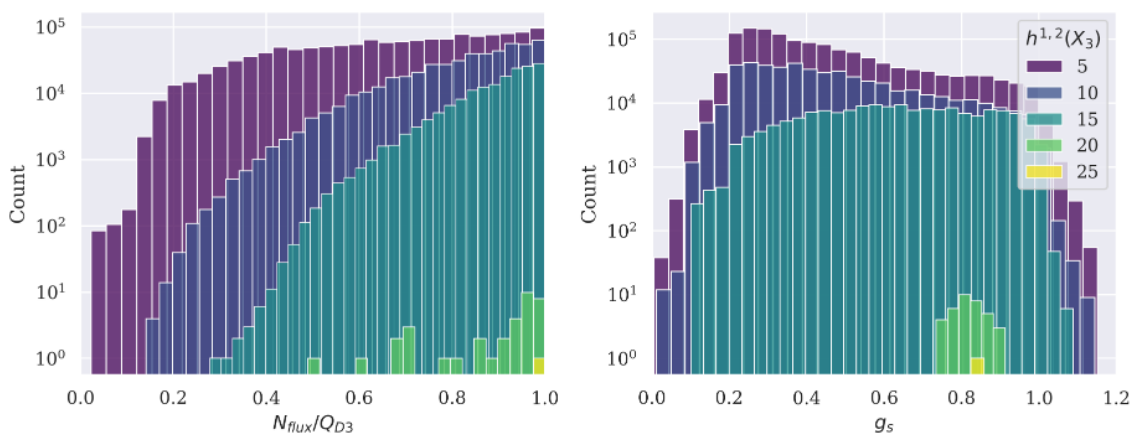
We present the performance of our code in figure 4. On the left, we show the time required to obtain roots to the  $F$ -term conditions by employing our optimisation module. Crucially, we observe virtually no scaling with respect to  $h^{1,2}$  which, as we said before, is largely due to `jit`-compilation.<sup>22</sup> This in principle opens up the opportunity to efficiently pursue systematic scans for flux vacua over a large range of models within the wider CY landscape.

On the right in figure 4, we plot the distributions of  $\varepsilon$  computed for instanton contributions up to degree 2 via eq. (2.18). We again find that our solutions have extremely small  $\varepsilon \lesssim 10^{-2}$  signifying excellent control over instanton corrections. Nonetheless, as we

<sup>22</sup>Recall that `vmap` is not being used currently in our optimisation module which would lead to a significant speed up over a few orders of magnitude. We however use `vmap` for all other parts of the algorithm e.g. when checking the validity of our solutions.



**Figure 4.** Efficiency analysis of our code for the models summarised in table 2: *Left*: time required to apply the optimisation module on a given number of input fluxes obtained from the average over five independent runs. *Right*: the distribution of the instanton control parameter  $\epsilon$  (recall eq. (2.18)). Note that all  $\epsilon$  values are far below our cut-off 0.1.



**Figure 5.** Solutions for large number of moduli. *Left*: distribution of the ratio of  $N_{\text{flux}}$  over the maximally allowed D3-charge  $Q_{D3}$ . *Right*: distribution of the string coupling  $g_s$ .

stressed several times before, a more thorough analysis is a prerequisite to make definitive statements about the radius of convergence of the sum over instanton terms.

The left plot in figure 5 shows the distribution of the ratio of  $N_{\text{flux}}$  over the maximally allowed D3-charge  $Q_{D3}$ . (Note that each sample at fixed  $h^{1,2}$  contains a different number of vacua as summarised in Tab 2.) As we argued before in section 4.1, our ISD<sub>+</sub> sampling, which we picked because of its reliability, prefers large  $N_{\text{flux}}$  which is why the distribution in figure 5 is peaked near  $N_{\text{flux}}/Q_{D3} \sim 1$ . In contrast,  $g_s$  is almost uniformly distributed in a wide range of values for  $h^{1,2} \leq 15$ , though the distribution for  $h^{1,2} = 5, 10$  exhibit moderate peaks around  $g_s \approx 0.2$ .

## 5 Conclusions

We presented an efficient and reliable framework to find flux vacua providing access to largely unexplored territories of string theory solutions. These advances have been achieved by combining various numerical techniques ubiquitously used in the context of machine learning, namely automatic differentiation, just-in-time compilation, and vectorisation. These methods allow for the generation of fast (C++), parallelisable (on CPU and GPU), and versatile code allowing a simple interface with EFT quantities, e.g., the couplings in the prepotential.

In this work, we are only hinting at upcoming physics analysis for these flux solutions. For instance, our framework, **JAXVacua**, intrinsically calculates the scalar potential and its derivatives at machine precision to any order in some approximation scheme. This enables systematic studies of properties of moduli potentials in general classes of models. Even more importantly, as we showed in the main text, large scale surveys of previously inaccessible regimes in the string landscape become feasible. This provides excellent opportunities to refine our understanding of the statistics of vacuum solutions, their associated mass spectra and couplings to visible matter fields. Ultimately, there is a wide range of applications of our methods to string model building and cosmology.

At this stage, there are a few technical improvements which we leave for the near future to make flux vacua at LCS of the entire KS database accessible:

- **Optimisation:** although we identify `scipy.optimize.root` as an efficient root finding method for flux vacua, it is nevertheless a limiting factor in efficiency. For instance, it would allow for further speed-up if we can use this method on the GPU directly. In this context it is interesting to see whether other optimisers, such as variants of gradient descent can lead to more efficient flux vacua generation, see [92] for applications of gradient descent methods to different types of string theory vacua. As the successful choice of optimiser is also based on the structure of the respective energy landscape in this optimisation problem, it will be interesting to compare the similarities between the string theory landscape and other energy landscapes (e.g. spin glass systems or deep learning optimisation).
- **Characterising the trustable EFT regime:** to understand the stability of flux solutions at the boundaries of the LCS expansion, efficient ways of estimating the radius of convergence in our examples are pending (see [16, 36, 61, 62] for examples). This applies to a subset of our solutions where the control parameter  $\mathcal{F}_{\text{inst}}/\mathcal{F}_{\text{pert}} = \varepsilon$  becomes large. Currently, we evaluate this parameter by calculating the GV expansion using **CYTools** [24] up to high degrees. In this way, we identified a large sample of solutions where these contributions are exponentially suppressed up to degree 10. Crucially, our implementation is able to compute the relevant quantities to any order in the GV expansion at machine precision.
- **Sampling input data:** we observed that the success rate of finding vacua below tadpole decreases in our current pipeline significantly at large  $h^{1,2} > 15$ . We believe

that this is mainly due to inefficient sampling procedures. To make progress, it would be instructive to formulate conditions that, for a given choice of fluxes, guarantee solutions to exist inside the Kähler cone. Secondly, while the ISD sampling methods have shown outstanding success, it turns out to be hard to sample points in the Kähler cone such that  $N_{\text{flux}} \leq Q_{D3}$ . By implementing this as an optimisation problem, gradient descent methods could help locating such points inside the Kähler cone.

We note that, although our methods and examples are currently limited to LCS limits, our implementation can be adapted to study different asymptotic limits in moduli space such as conifold regimes [63, 64]. In these cases, the solutions to the Picard-Fuchs equation for the periods have different analytic properties. Such generalisations will be investigated in the future.

More generally speaking, our approach in JAXVacua can be extended to other moduli, such as Kähler moduli stabilisation. This will enable a general search for vacua where the presence of novel minima could be observed that fall neither in the class of the standard KKLT nor the LVS. Such new solutions are believed to arise from the intricate structure of EFTs from string compactifications. In the presence of large numbers of moduli, such regions are most easily discovered in comprehensive numerical explorations as presented in this paper.

## Acknowledgments

We would like to thank Alex Cole, Mathis Gerdes, Arthur Hebecker, Manki Kim, Severin Lust, Liam McAllister, Jakob Moritz, Richard Nally and Gary Shiu for useful discussions. We especially thank Andres Rios-Tascon for providing the code to compute GV and GW invariants. AS thanks DAMTP at the University of Cambridge and Ludwig Maximilian University of Munich for hospitality where parts of this work have been completed. The research of AS is supported by NSF grant PHY-2014071.

## A Implementation details of JAXVacua

In this appendix we provide some more details on the implementation. A public release of this code is intended in the future when we have developed a few more modules. In the meantime, the following discussion might prove useful in the implementation of similar optimisation problems.

- As described above (recall (2.14)), the prepotential can be implemented as

```

1 import jax.numpy as jnp
2 from jax import jit
3 from scipy.special import zeta
4
5 @jit
6 def prepotential(moduli):
```

```

7     return jnp.einsum('ijk,i,j,k',intnums,moduli,moduli,
    moduli)/6. + jnp.einsum('ij,i,j',a_matrix,moduli,moduli
    )/2. +jnp.einsum('i,i',b_vector,moduli) + zeta(3)*chi
    /(2.*jnp.pi*1j)**(3)/2.-jnp.sum(GW_inv*jnp.exp(2.*jnp.
    pi*1j*jnp.einsum("ki,i",curve_charges,moduli)))/(2.*jnp
    .pi*1j)**(3)

```

Here, we use `jit` to speed up the valuation time of the function `prepotential(moduli)`. The invariants like  $\kappa_{ijk}$  (called `intnums` in the code snippet above) are globally defined variables.

- As described in the main text, many functions require the combination of various gradients of functions. Such gradients are implemented using `jax.grad`. For instance, the gradient of the above prepotential is easily obtained by calling `jax.grad`, i.e.,

```

1 @jit
2 def gradient_prepotential(moduli):
3     return jax.grad(prepotential, holomorphic=True)(moduli)

```

We use the optional `holomorphic=True` argument because the prepotential is a holomorphic and, in particular, complex function.

- We use `scipy.optimize.root` to find solutions to  $D_I W = 0$ . The input is the starting point for the moduli is `x` and `flux` is the flux vector, leading to an implementation as follows

```

1 from scipy.optimize import root
2
3 def sciRoot(x, flux):
4     res = root(DW, x0 = x, args = (flux, ), tol = 1e-10)
5     return [res.x , res.success]

```

The return values are the candidate roots `res.x` and the status of the root finder `res.success`. If the latter indicates a failure, we discard the flux choice.

- The computation of some quantities, such as the Hessian of the potential, can be significantly speed up using the JAX vectorisation functionality `jax.vmap`. For example, to compute the Hessian on all of our solutions, we achieve this as follows:

```

1 from jax import vmap
2
3 hessian_vmap = vmap(lambda moduli, conj_moduli, tau,
    conj_tau, fluxes: hessian(moduli, conj_moduli, tau,
    conj_tau, fluxes, mode="SUSY"))

```

Here, setting `mode="SUSY"` assumes  $D_I W = 0$  for all fields, thereby speeding up the calculation of the Hessian at the minimum.

**Open Access.** This article is distributed under the terms of the Creative Commons Attribution License ([CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)), which permits any use, distribution and reproduction in any medium, provided the original author(s) and source are credited.

## References

- [1] S. Ashok and M.R. Douglas, *Counting flux vacua*, *JHEP* **01** (2004) 060 [[hep-th/0307049](#)] [[INSPIRE](#)].
- [2] F. Denef and M.R. Douglas, *Distributions of flux vacua*, *JHEP* **05** (2004) 072 [[hep-th/0404116](#)] [[INSPIRE](#)].
- [3] W. Taylor and Y.-N. Wang, *The F-theory geometry with most flux vacua*, *JHEP* **12** (2015) 164 [[arXiv:1511.03209](#)] [[INSPIRE](#)].
- [4] W. Lerche, D. Lust and A.N. Schellekens, *Chiral four-dimensional heterotic strings from selfdual lattices*, *Nucl. Phys. B* **287** (1987) 477 [[INSPIRE](#)].
- [5] D. Baumann and L. McAllister, *Inflation and string theory*, Cambridge University Press, Cambridge, U.K. (2015) [[DOI:10.1017/CB09781316105733](#)] [[INSPIRE](#)].
- [6] M. Cicoli et al., *String cosmology: from the early universe to today*, [arXiv:2303.04819](#) [[INSPIRE](#)].
- [7] M. Kreuzer and H. Skarke, *Complete classification of reflexive polyhedra in four-dimensions*, *Adv. Theor. Math. Phys.* **4** (2000) 1209 [[hep-th/0002240](#)] [[INSPIRE](#)].
- [8] F. Denef, M.R. Douglas and B. Florea, *Building a better racetrack*, *JHEP* **06** (2004) 034 [[hep-th/0404257](#)] [[INSPIRE](#)].
- [9] R. Blumenhagen, V. Braun, T.W. Grimm and T. Weigand, *GUTs in type IIB orientifold compactifications*, *Nucl. Phys. B* **815** (2009) 1 [[arXiv:0811.2936](#)] [[INSPIRE](#)].
- [10] M. Cicoli, C. Mayrhofer and R. Valandro, *Moduli stabilisation for chiral global models*, *JHEP* **02** (2012) 062 [[arXiv:1110.3333](#)] [[INSPIRE](#)].
- [11] M. Cicoli et al., *D-branes at del Pezzo singularities: global embedding and moduli stabilisation*, *JHEP* **09** (2012) 019 [[arXiv:1206.5237](#)] [[INSPIRE](#)].
- [12] M. Demirtas, A. Rios-Tascon and L. McAllister, *CYTools: a software package for analyzing Calabi-Yau manifolds*, [arXiv:2211.03823](#) [[INSPIRE](#)].
- [13] P. Jefferson and M. Kim, *On the intermediate Jacobian of M5-branes*, [arXiv:2211.00210](#) [[INSPIRE](#)].
- [14] J. Moritz, *Orientifolding Kreuzer-Skarke*, [arXiv:2305.06363](#) [[INSPIRE](#)].
- [15] P. Candelas, X.C. De La Ossa, P.S. Green and L. Parkes, *A pair of Calabi-Yau manifolds as an exactly soluble superconformal theory*, *Nucl. Phys. B* **359** (1991) 21 [[INSPIRE](#)].
- [16] S. Hosono, A. Klemm, S. Theisen and S.-T. Yau, *Mirror symmetry, mirror map and applications to Calabi-Yau hypersurfaces*, *Commun. Math. Phys.* **167** (1995) 301 [[hep-th/9308122](#)] [[INSPIRE](#)].
- [17] S. Gukov, C. Vafa and E. Witten, *CFT's from Calabi-Yau four folds*, *Nucl. Phys. B* **584** (2000) 69 [[hep-th/9906070](#)] [[INSPIRE](#)].
- [18] S.B. Giddings, S. Kachru and J. Polchinski, *Hierarchies from fluxes in string compactifications*, *Phys. Rev. D* **66** (2002) 106006 [[hep-th/0105097](#)] [[INSPIRE](#)].

- [19] R. Blumenhagen, M. Cvetič, S. Kachru and T. Weigand, *D-brane instantons in type II orientifolds*, *Ann. Rev. Nucl. Part. Sci.* **59** (2009) 269 [[arXiv:0902.3251](#)] [[INSPIRE](#)].
- [20] M. Demirtas et al., *Small cosmological constants in string theory*, *JHEP* **12** (2021) 136 [[arXiv:2107.09064](#)] [[INSPIRE](#)].
- [21] S. Alexandrov et al., *D-instanton induced superpotential*, *JHEP* **07** (2022) 090 [[arXiv:2204.02981](#)] [[INSPIRE](#)].
- [22] N. Gendler et al., *Superpotentials from singular divisors*, *JHEP* **11** (2022) 142 [[arXiv:2204.06566](#)] [[INSPIRE](#)].
- [23] M. Kim, *On D3-brane superpotential*, [arXiv:2207.01440](#) [[INSPIRE](#)].
- [24] M. Demirtas et al., *Computational mirror symmetry*, [arXiv:2303.00757](#) [[INSPIRE](#)].
- [25] M. Kim, *D-instanton, threshold corrections, and topological string*, *JHEP* **05** (2023) 097 [[arXiv:2301.03602](#)] [[INSPIRE](#)].
- [26] M. Kim, *On string one-loop correction to the Einstein-Hilbert term and its implications on the Kähler potential*, *JHEP* **07** (2023) 044 [[arXiv:2302.12117](#)] [[INSPIRE](#)].
- [27] M. Kim, *On one-loop corrected dilaton action in string theory*, [arXiv:2305.08263](#) [[INSPIRE](#)].
- [28] S. Kachru, R. Kallosh, A.D. Linde and S.P. Trivedi, *De Sitter vacua in string theory*, *Phys. Rev. D* **68** (2003) 046005 [[hep-th/0301240](#)] [[INSPIRE](#)].
- [29] V. Balasubramanian, P. Berglund, J.P. Conlon and F. Quevedo, *Systematics of moduli stabilisation in Calabi-Yau flux compactifications*, *JHEP* **03** (2005) 007 [[hep-th/0502058](#)] [[INSPIRE](#)].
- [30] I. Garcia-Etxebarria, H. Hayashi, R. Savelli and G. Shiu, *On quantum corrected Kahler potentials in F-theory*, *JHEP* **03** (2013) 005 [[arXiv:1212.4831](#)] [[INSPIRE](#)].
- [31] R. Minasian, T.G. Pugh and R. Savelli, *F-theory at order  $\alpha'^3$* , *JHEP* **10** (2015) 050 [[arXiv:1506.06756](#)] [[INSPIRE](#)].
- [32] D. Klaewer, S.-J. Lee, T. Weigand and M. Wiesner, *Quantum corrections in 4d  $N = 1$  infinite distance limits and the weak gravity conjecture*, *JHEP* **03** (2021) 252 [[arXiv:2011.00024](#)] [[INSPIRE](#)].
- [33] M. Cicoli et al., *Systematics of the  $\alpha'$  expansion in F-theory*, *JHEP* **08** (2021) 099 [[arXiv:2106.04592](#)] [[INSPIRE](#)].
- [34] D. Martinez-Pedraza, D. Mehta, M. Rummel and A. Westphal, *Finding all flux vacua in an explicit example*, *JHEP* **06** (2013) 110 [[arXiv:1212.4530](#)] [[INSPIRE](#)].
- [35] M. Cicoli et al., *Explicit de Sitter flux vacua for global string models with chiral matter*, *JHEP* **05** (2014) 001 [[arXiv:1312.0014](#)] [[INSPIRE](#)].
- [36] S. Hosono, A. Klemm, S. Theisen and S.-T. Yau, *Mirror symmetry, mirror map and applications to complete intersection Calabi-Yau spaces*, *Nucl. Phys. B* **433** (1995) 501 [[hep-th/9406055](#)] [[INSPIRE](#)].
- [37] D.A. Cox and S. Katz, *Mirror symmetry and algebraic geometry*, American Mathematical Society, U.S.A. (2000) [[INSPIRE](#)].
- [38] J. Bradbury et al., *JAX: composable transformations of Python+NumPy programs*, <http://github.com/google/jax>.

- [39] A. Cole, A. Schachner and G. Shiu, *Searching the landscape of flux vacua with genetic algorithms*, *JHEP* **11** (2019) 045 [[arXiv:1907.10072](#)] [[INSPIRE](#)].
- [40] S. Krippendorff, R. Kroepsch and M. Syvaeri, *Revealing systematics in phenomenologically viable flux vacua with reinforcement learning*, [arXiv:2107.04039](#) [[INSPIRE](#)].
- [41] A. Cole, S. Krippendorff, A. Schachner and G. Shiu, *Probing the structure of string theory vacua with genetic algorithms and reinforcement learning*, in the proceedings of the 35<sup>th</sup> conference on neural information processing systems, (2021) [[arXiv:2111.11466](#)] [[INSPIRE](#)].
- [42] S. Krippendorff and V.V. Camell, *Towards structures in the flux landscape at large number of moduli*, [arXiv:2209.15443](#) [[INSPIRE](#)].
- [43] M. Grana, *Flux compactifications in string theory: a comprehensive review*, *Phys. Rept.* **423** (2006) 91 [[hep-th/0509003](#)] [[INSPIRE](#)].
- [44] M.R. Douglas and S. Kachru, *Flux compactification*, *Rev. Mod. Phys.* **79** (2007) 733 [[hep-th/0610102](#)] [[INSPIRE](#)].
- [45] A. Hebecker, *Naturalness, string landscape and multiverse: a modern introduction with exercises*, *Lect. Notes Phys.* **979** (2021) 1 [[INSPIRE](#)].
- [46] F. Denef, *Lectures on constructing string vacua*, *Les Houches* **87** (2008) 483 [[arXiv:0803.1194](#)] [[INSPIRE](#)].
- [47] T.W. Grimm, E. Plauschinn and D. van de Heisteeg, *Moduli stabilization in asymptotic flux compactifications*, *JHEP* **03** (2022) 117 [[arXiv:2110.05511](#)] [[INSPIRE](#)].
- [48] S. Kachru, M. Kim, L. Mcallister and M. Zimet, *De Sitter vacua from ten dimensions*, *JHEP* **12** (2021) 111 [[arXiv:1908.04788](#)] [[INSPIRE](#)].
- [49] M. Demirtas, M. Kim, L. Mcallister and J. Moritz, *Vacua with small flux superpotential*, *Phys. Rev. Lett.* **124** (2020) 211603 [[arXiv:1912.10047](#)] [[INSPIRE](#)].
- [50] C. Brodie and M.C.D. Marsh, *The spectra of type IIB flux compactifications at large complex structure*, *JHEP* **01** (2016) 037 [[arXiv:1509.06761](#)] [[INSPIRE](#)].
- [51] M.C.D. Marsh and K. Sousa, *Universal properties of type IIB and F-theory flux compactifications at large complex structure*, *JHEP* **03** (2016) 064 [[arXiv:1512.08549](#)] [[INSPIRE](#)].
- [52] J.J. Blanco-Pillado, K. Sousa, M.A. Urkiola and J.M. Wachter, *Universal class of type-IIB flux vacua with analytic mass spectrum*, *Phys. Rev. D* **103** (2021) 106006 [[arXiv:2011.13953](#)] [[INSPIRE](#)].
- [53] F. Marchesano, D. Prieto and M. Wiesner, *F-theory flux vacua at large complex structure*, *JHEP* **08** (2021) 077 [[arXiv:2105.09326](#)] [[INSPIRE](#)].
- [54] T. Coudarchet, F. Marchesano, D. Prieto and M.A. Urkiola, *Analytics of type IIB flux vacua and their mass spectra*, *JHEP* **01** (2023) 152 [[arXiv:2212.02533](#)] [[INSPIRE](#)].
- [55] H. Jockers et al., *Two-sphere partition functions and Gromov-Witten invariants*, *Commun. Math. Phys.* **325** (2014) 1139 [[arXiv:1208.6244](#)] [[INSPIRE](#)].
- [56] B. Bastian, T.W. Grimm and D. van de Heisteeg, *Modeling general asymptotic Calabi-Yau periods*, [arXiv:2105.02232](#) [[INSPIRE](#)].
- [57] D.R. Morrison, *Picard-Fuchs equations and mirror maps for hypersurfaces*, *AMS/IP Stud. Adv. Math.* **9** (1998) 185 [[hep-th/9111025](#)] [[INSPIRE](#)].

- [58] S. Hosono, A. Klemm and S. Theisen, *Lectures on mirror symmetry*, *Lect. Notes Phys.* **436** (1994) 235 [[hep-th/9403096](#)] [[INSPIRE](#)].
- [59] R. Gopakumar and C. Vafa, *M theory and topological strings*. 1, [hep-th/9809187](#) [[INSPIRE](#)].
- [60] R. Gopakumar and C. Vafa, *M theory and topological strings*. 2, [hep-th/9812127](#) [[INSPIRE](#)].
- [61] P. Candelas, A. Font, S.H. Katz and D.R. Morrison, *Mirror symmetry for two parameter models*. 2, *Nucl. Phys. B* **429** (1994) 626 [[hep-th/9403187](#)] [[INSPIRE](#)].
- [62] A. Klemm and E. Zaslow, *Local mirror symmetry at higher genus*, *AMS/IP Stud. Adv. Math.* **23** (2001) 183 [[hep-th/9906046](#)] [[INSPIRE](#)].
- [63] M. Demirtas, M. Kim, L. McAllister and J. Moritz, *Conifold vacua with small flux superpotential*, *Fortsch. Phys.* **68** (2020) 2000085 [[arXiv:2009.03312](#)] [[INSPIRE](#)].
- [64] R. Álvarez-García, R. Blumenhagen, M. Brinkmann and L. Schlechter, *Small flux superpotentials for type IIB flux vacua close to a conifold*, *Fortsch. Phys.* **68** (2020) 2000088 [[arXiv:2009.03325](#)] [[INSPIRE](#)].
- [65] V.V. Batyrev, *Dual polyhedra and mirror symmetry for Calabi-Yau hypersurfaces in toric varieties*, *J. Alg. Geom.* **3** (1994) 493 [[alg-geom/9310003](#)] [[INSPIRE](#)].
- [66] M. Demirtas, L. McAllister and A. Rios-Tascon, *Bounding the Kreuzer-Skarke landscape*, *Fortsch. Phys.* **68** (2020) 2000086 [[arXiv:2008.01730](#)] [[INSPIRE](#)].
- [67] C.T.C. Wall, *Classification problems in differential topology*. V, *Invent. Math.* **1** (1966) 355 [[INSPIRE](#)].
- [68] P. Candelas and X.C. de la Ossa, *Comments on conifolds*, *Nucl. Phys. B* **342** (1990) 246 [[INSPIRE](#)].
- [69] F. Carta, J. Moritz and A. Westphal, *A landscape of orientifold vacua*, *JHEP* **05** (2020) 107 [[arXiv:2003.04902](#)] [[INSPIRE](#)].
- [70] T.W. Grimm, *Taming the landscape of effective theories*, *JHEP* **11** (2022) 003 [[arXiv:2112.08383](#)] [[INSPIRE](#)].
- [71] J.P. Conlon and F. Quevedo, *On the explicit construction and statistics of Calabi-Yau flux vacua*, *JHEP* **10** (2004) 039 [[hep-th/0409215](#)] [[INSPIRE](#)].
- [72] M. Demirtas, C. Long, L. McAllister and M. Stillman, *The Kreuzer-Skarke axiverse*, *JHEP* **04** (2020) 138 [[arXiv:1808.01282](#)] [[INSPIRE](#)].
- [73] N. Gendler et al., *Moduli space reconstruction and weak gravity*, [arXiv:2212.10573](#) [[INSPIRE](#)].
- [74] P. Candelas, A.M. Dale, C.A. Lutken and R. Schimmrigk, *Complete intersection Calabi-Yau manifolds*, *Nucl. Phys. B* **298** (1988) 493 [[INSPIRE](#)].
- [75] L.B. Anderson, X. Gao, J. Gray and S.-J. Lee, *Fibrations in CICY threefolds*, *JHEP* **10** (2017) 077 [[arXiv:1708.07907](#)] [[INSPIRE](#)].
- [76] M. Abadi et al., *TensorFlow: large-scale machine learning on heterogeneous systems*, <https://www.tensorflow.org/>.
- [77] S. de Alwis et al., *Moduli spaces in AdS<sub>4</sub> supergravity*, *JHEP* **05** (2014) 102 [[arXiv:1312.5659](#)] [[INSPIRE](#)].
- [78] P. Virtanen et al., *SciPy 1.0 — fundamental algorithms for scientific computing in Python*, *Nature Meth.* **17** (2020) 261 [[arXiv:1907.10121](#)] [[INSPIRE](#)].

- [79] M.J.D. Powell, *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, *Comput. J.* **7** (1964) 155 [[INSPIRE](#)].
- [80] O. DeWolfe, A. Giryavets, S. Kachru and W. Taylor, *Enumerating flux vacua with enhanced symmetries*, *JHEP* **02** (2005) 037 [[hep-th/0411061](#)] [[INSPIRE](#)].
- [81] D. Gallego, M.C.D. Marsh, B. Vercnocke and T. Wrase, *A new class of de Sitter vacua in type IIB large volume compactifications*, *JHEP* **10** (2017) 193 [[arXiv:1707.01095](#)] [[INSPIRE](#)].
- [82] C. Long, L. McAllister and J. Stout, *Systematics of axion inflation in Calabi-Yau hypersurfaces*, *JHEP* **02** (2017) 014 [[arXiv:1603.01259](#)] [[INSPIRE](#)].
- [83] K. Tsagkaris and E. Plauschinn, *Moduli stabilization in type IIB orientifolds at  $h^{2,1} = 50$* , *JHEP* **03** (2023) 049 [[arXiv:2207.13721](#)] [[INSPIRE](#)].
- [84] D. Marsh, L. McAllister and T. Wrase, *The wasteland of random supergravities*, *JHEP* **03** (2012) 102 [[arXiv:1112.3034](#)] [[INSPIRE](#)].
- [85] T.C. Bachlechner, D. Marsh, L. McAllister and T. Wrase, *Supersymmetric vacua in random supergravity*, *JHEP* **01** (2013) 136 [[arXiv:1207.2763](#)] [[INSPIRE](#)].
- [86] A. Giryavets, S. Kachru, P.K. Tripathy and S.P. Trivedi, *Flux compactifications on Calabi-Yau threefolds*, *JHEP* **04** (2004) 003 [[hep-th/0312104](#)] [[INSPIRE](#)].
- [87] J. Louis, M. Rummel, R. Valandro and A. Westphal, *Building an explicit de Sitter*, *JHEP* **10** (2012) 163 [[arXiv:1208.3208](#)] [[INSPIRE](#)].
- [88] D.J. Bates, D. Brake and M. Niemerg, *Paramotopy: parameter homotopies in parallel*, [arXiv:1804.04183](#).
- [89] N. Clerc and A. Finoguenov, *X-ray cluster cosmology*, in *Handbook of X-ray and gamma-ray astrophysics*, Springer, Singapore (2023), p. 1 [[DOI:10.1007/978-981-16-4544-0\\_117-1](#)] [[arXiv:2203.11906](#)] [[INSPIRE](#)].
- [90] C. Crinò, F. Quevedo, A. Schachner and R. Valandro, *A database of Calabi-Yau orientifolds and the size of D3-tadpoles*, *JHEP* **08** (2022) 050 [[arXiv:2204.13115](#)] [[INSPIRE](#)].
- [91] B.R. Greene and M.R. Plesser, *Duality in Calabi-Yau moduli space*, *Nucl. Phys. B* **338** (1990) 15 [[INSPIRE](#)].
- [92] I.M. Comsa, M. Firsching and T. Fischbacher, *SO(8) supergravity and the magic of machine learning*, *JHEP* **08** (2019) 057 [[arXiv:1906.00207](#)] [[INSPIRE](#)].