ORIGINAL PAPER

Fortschritte der Physik
Progress of Physics

www.fp-journal.org

# Statistical Predictions in String Theory and Deep Generative Models

*James Halverson* and Cody Long*

**Generative models in deep learning allow for sampling probability distributions that approximate data distributions. We propose using generative models for making approximate statistical predictions in the string theory landscape. For vacua admitting a Lagrangian description this can be thought of as learning random tensor approximations of couplings. As a concrete proof-of-principle, we demonstrate in a large ensemble of Calabi-Yau manifolds that Kähler metrics evaluated at points in Kähler moduli space are well-approximated by ensembles of matrices produced by a deep convolutional Wasserstein GAN. Accurate approximations of the Kähler metric eigenspectra are achieved with far fewer than $h^{11}$ Gaussian draws. Accurate extrapolation to values of $h^{11}$ outside the training set are achieved via a conditional GAN. Together, these results implicitly suggest the existence of strong correlations in the data, as might be expected if Reid's fantasy is correct.**

## 1. Introduction

String theory is a leading candidate for unifying quantum gravity with particle physics and cosmology. It has a large landscape of vacua, due not only to the plethora of fluxes[1–4] that may exist in its extra dimensions, but also the large number of extra-dimensional geometries[5–7] themselves; the known size of both has grown significantly in recent years. The landscape gives rise to rich cosmological dynamics and diverse low energy compactifications that may exhibit features of the Standard Models of particle physics and cosmology, as well as observable remnants of the ultraviolet theory. If string theory is true, fundamental physics is a complex system.

The diversity of possibilities has a simple implication: predictions in string theory are statistical.[8] Given the distribution $P(i)$ on vacua, one would like to compute expectation values of observables $\mathcal{O}$

$$\mathbb{E}_{i\sim P(i)}[\mathcal{O}] = \sum_{i\in S_{\rm vac}} D(i)A(i)\ \mathcal{O}(i), \tag{1}$$

where we have written $P(i) = D(i)A(i)$ in a factorized form involving a dynamical factor $D(i)$ and an anthropic factor $A(i)$. These factors give important corrections from a naive uniform distribution. Unfortunately, neither the full set of vacua $S_{\rm vac}$ nor the factors $D(i)$ or $A(i)$ are currently known in full, and significant theoretical work is required to determine them. However, when drawing from a uniform distribution, the largest known sets of flux vacua[4] and geometries[6,7] both suggest that large numbers of gauge sectors and axion-like particles are the rule, not the exception; both have significant cosmological implications, see, e.g., [9–11]. A number of proposals exist for the dynamical factor, including global measures,[12,13] local measures,[14–17] and computational complexity based measures.[18–20] It is even more difficult to compute $A(i)$, butthere is significant evidence that it depends on the cosmological constant.[1,21]

Though the full extent to which computational complexity affects the dynamical factor is not known, it certainly affects practical efforts to study the landscape. That is, difficulties arises not only from the large number of vacua, but also the computational complexity of physical questions related to them. For instance, finding small cosmological constants in the Bousso-Polchinski model is NP-complete,[22] solving decision problems in the landscape runs up against Diophantine undecidability,[23] and both constructing and minimizing the scalar potential requires[24] solving instances of NP-hard and co-NP-hard problems. In some cases the structure of the theory may allow for the avoidance of worst-case complexity, for instance in ED3-instanton problems,[25] but identifying such instances can itself be challenging.

Alternatively, approximations can allow for the avoidance of complexity. For instance, so-called fully-polynomial time approximation schemes are algorithms for solving a problem with error bounded by $\epsilon$, such that the runtime is polynomial in the input size and $1/\epsilon$, even if the exact version of the problem is NP-hard.

Complexity provides a significant obstacle to making statistical predictions in string compactifications. As such, it is natural to wonder whether one could get by with making approximate predictions, using appropriate approximations

$$\hat{S}_{\rm vac} \simeq S_{\rm vac}, \qquad \hat{D}(i) \simeq D(i), \qquad \hat{A}(i) \simeq A(i). \tag{2}$$

Rather than computing expectation values of observables in an exact distribution on string vacua, one could attempt to make predictions in an approximate distribution. Put differently, if exact

J. Halverson, C. Long
Department of Physics
Northeastern University
Boston, MA 02115-5000, USA
E-mail: j.halverson@neu.edu

DOI: 10.1002/prop.202000005

 

calculations in string theory are too slow, could fast-but-accurate simulation suffice?

Determining approximations to distributions of data $P_d$ is the subject of so-called generative models in deep learning.[1] In generative models, a random variable $z$ drawn from $P(z)$, written $z \sim P(z)$, is passed through a parameterized function from a latent space $\mathcal{Z}$ to a data space $\mathcal{D}$,

$$F_\theta : \mathcal{Z} \to \mathcal{D}, \tag{3}$$

generating a sample $F_\theta(z)$ of an implicit distribution $P_\theta$ that depends on the parameters $\theta$ in $F_\theta$; often, $F_\theta$ is a deep neural network. There are many classes of generative models, corresponding to different functional forms for $F_\theta$ and different algorithms for optimizing its parameters. The optimization procedure leads to increasingly good approximations $P_\theta \simeq P_d$, as measured by an appropriate distance measure such as the Kullback-Leibler divergence or the Wasserstein distance.

In this paper we explore utilizing generative models to make statistical predictions in string theory.

This is a rather general idea, but it is very concrete for vacua admitting a Lagrangian description, where modeling statistical predictions involves learning random tensor approximations of the couplings in the Lagrangian. For matrix couplings, such as mass matrices or metrics appearing in kinetic terms, this amounts to directly learning a random matrix ensemble that simulates string data, rather than attempting to guess one a priori.

As a direct application, we use a class of generative models known as generative adversarial networks (GANs)[32] to learn a random matrix approximation to Kähler metrics evaluated at points on the Kähler moduli space of Calabi-Yau manifolds.[2] In each case, a generator neural network $G_\theta$ is trained on Kähler metrics obtained in Calabi-Yau compactification by optimization of the parameters $\theta$. For any epoch in the training, $G_\theta$ can be used to generate simulated Kähler metrics that model real ones increasingly well as training proceeds, as measured for instance by the Wasserstein distance on the log eigenspectra of the real and simulated Kähler metrics.

For those less familiar with generative models, we would like to highlight the important role of the noise $z \sim P(z)$, which has dimension $n_z$, used in simulation. The training process involves optimizing the network $G_\theta$ so that noise sent through the network models data, where in general one expects that the value of $n_z$ affects the quality of the model, i.e. there is some minimal dimension of input necessary to model the data. In models where GANs are trained at fixed values of $h^{11}$, we find that performance is relatively insensitive to $n_z$, provided $n_z \gtrsim 5$, which itself is much below $h^{11}$. This implicit suggests the existence of correlations in the data.

Since many string vacua arise at large[3] $N$, where calculations are often intractable, it would be useful to have a fast-but-accurate simulator of string data in that regime. This requires extrapolation outside of the training sample, which a priori is difficult unless significant structure exists in the data that allows for extrapolation. We demonstrate that a conditional GAN gives rise to better-than-expected extrapolation to larger values of $h^{11}$ for Kähler metrics.

This paper is organized as follows. In Section 2 we review generative models and introduce the original GAN and the Wasserstein GAN. In Section 3 we introduce the use of generative models for learning random tensor approximations of string effective Lagrangians, exemplifying the idea for Kähler metrics at fixed $h^{11}$ in Section 3.1 and extrapolating to $h^{11}$ values outside the training set in Section 3.2. In Section 4 we review the main results. That section is the primary location where results are discussed, in an effort to separate the implications of the results from somewhat technical deep learning details utilized in their derivation.

## 2. Generative Models

We have already introduced the essential idea behind generative models: to learn how to generate samples of a distribution that closely approximates a data distribution, i.e., to produce reliable (and fast) simulations.

A myriad of generative models exist. Common models not utilized in this work include variational autoencoders[34] and normalizing flows.[35] The former provide a modification of an autoencoder architecture such that the second half of a network may be used to generate data given draws from a multivariate Gaussian, while the latter focuses on utilizing an invertible architecture, which in turn allows for the evaluation of sample probabilities via inversion. Potential uses of these techniques in string theory will be discussed in Section 4.

The generative models that we utilize in this work are generative adversarial networks[32] (GANs). GANs pit a generator network against an adversary network, often referred to as a discriminator or critic, where the training goal of the generator is to produce fake samples from noise that fool the adversary, while the latter discriminator is trained to determine whether the samples it sees are real or were produced by the generator. We will utilize a variety of GANs, which differ according to their loss functions[32,36] and network architecture.[37] For the purposes of interpolation and extrapolation, we will also pass conditions to the GAN,[38] where in our case the condition will be a value of $h^{11}$ for which to simulate a Kähler metric. In the end, we will find that a Wasserstein GAN with deep convolutional architecture outperforms the others.[4]

Let us review the original GAN[32] as we utilize it in this paper. It consists of a generator and a discriminator network

$$G_\theta : \mathbb{R}^{n_z} \to \mathbb{R}^N \times \mathbb{R}^N$$
$$D_w : \mathbb{R}^N \times \mathbb{R}^N \to [0, 1], \tag{4}$$

parameterized by $\theta$ and $w$, respectively. Real and fake data are labelled 1 and 0, respectively. We will sometimes abbreviate $G_\theta$

[1] Applications of deep learning to string theory have been of recent interest. See [26–29] for original works,[7,13,30] for further progress with a variety of techniques, and [31] for a review.

[2] See [33] for a study of the use of GANs in generating EFTs.

[3] Throughout, when we are vague about $N$ in string theory it is a proxy for the number of degrees of freedom, fluxes, cycles, etc.

[4] A comparative study[39] of the performance of different GANs suggests that fine-tuning of hyperparameters can sometimes compensate for fundamental algorithmic differences.

and $D_w$ to $G$ and $D$. During training, the generator is trained on a batch of simulated data $G(z)$ generated from a batch of noise $z \sim P(z)$ drawn from a noise distribution. The discriminator is trained on equal-size batches of real data $x \sim P_d(x)$ and simulated data $G(z)$. From the batches, the parameters are updated according to the loss functions

$$L_D^{\mathrm{GAN}} = -\mathbb{E}_{x\sim P_d(x)}[log(D(x)] + \mathbb{E}_{z\sim P(z)}[log(1 - D(G(z)))]$$
$$L_G^{\mathrm{GAN}} = -\mathbb{E}_{z\sim P(z)}[log(D(G(z)))]. \quad (5)$$

These losses may be interpreted term by term. For instance, consider a given $z \sim P(z)$ such that the discriminator thinks the generated data is real, i.e. $D(G(z)) = 1$. In this case the simulated $G(z)$ does not contribute to $L_G^{\mathrm{GAN}}$ but gives a large contribution to $L_D^{\mathrm{GAN}}$, which is as desired since the generator has fooled the discriminator into thinking that $G(z)$ is real. The converse holds as expected if $D(G(z)) = 0$, the generator is penalized since the discriminator has detected that $G(z)$ is a fake. Similarly, real data $x \sim P_d(x)$ penalizes $D$ by a positive contribution to $L_D^{\mathrm{GAN}}$ when $D(x) < 0$ i.e. when the discriminator is not sure that $x$ is real.

We also utilize Wasserstein GANs (WGANs),[36] which differ from the original GAN in important ways. There is an intuitive understanding and a more formal one; we begin by describing the latter, which will lead to an intuitive understanding after an approximation.

The WGAN is built on a solid theoretical foundation. Following,[36] consider ways in which to measure how close the model distribution $P_\theta$ is to the real distribution $P_d$, as measured by $\rho(P_\theta, P_d)$; in some cases $\rho$ is a proper distance, but in other often-used cases it is a divergence (such as the Kullback-Leibler divergence) that is not symmetric in the two distributions. A sequence of distributions $P_t$ with $t \in \mathbb{N}$ is said to converge (in $\rho$) to a distribution $P_\infty$ if $\rho(P_t, P_\infty)$ goes to zero as $t \to \infty$. Given two distances or divergences $\rho$ and $\rho'$, if the set of sequences convergent under $\rho$ is a superset of those convergent under $\rho'$, then it is said that $\rho$ induces a weaker topology; that is, has better convergence properties.

Since a GAN learns an implicit probability distribution, a natural learning question is which distance or divergence has the weakest topology, i.e. will have the best convergence properties to the data distribution. Four possibilities are considered in [36], according to whether $\rho$ is the total variation (TV) distance, the Kullback-Leibler (KL) divergence, the Jensen-Shannon (JS) divergence, or earth-mover (EM) distance, which is also known as the Wasserstein distance. The main theorem in [36] shows that the Wasserstein distance has the best convergence properties, followed by JS and TV, followed by KL, suggesting the utilizing the Wasserstein distance in a GAN could lead to superior training.

Unfortunately, the Wasserstein distance

$$W(P_\theta, P_d) = \inf_{\gamma\in\Pi(P_d,P_\theta)} \mathbb{E}_{(x,y)\sim\gamma}[\,||x - y||\,], \quad (6)$$

is intractable to compute for high dimensional distributions. Here $\Pi(P_d, P_\theta)$ are all joint distributions whose marginals are $P_d$ and $P_\theta$. However, Kantorovich-Rubinstein duality (see, e.g., [40]) allows it to be rewritten as

$$W(P_d, P_\theta) = \frac{1}{K} \sup_{||f||_L\le K} \mathbb{E}_{x\sim P_d}[f(x)] - \mathbb{E}_{y\sim P_\theta}[f(y)], \quad (7)$$

which involves the supremum over all functions $f$ to $\mathbb{R}$ that are $K$-Lipschitz for some constant $K$, i.e. $|f(a) - f(b)| \le K \times |a - b|$ for all $a, b$ on the domain, denoted $||f||_L \le K$. One could instead consider maximizing over a family of functions $f_w$ parameterized by $w \in W$, where compact $W$ ensures that $f_w$ is $K$-Lipschitz for some $K$. For instance $f_w$ could be a neural network with weights clamped to a compact space.

With this introduction, we can now reintroduce the generator $G_\theta$. Let $P_d$ again be the data distribution, and $P_\theta$ be the implicit distribution of $G_\theta(z)$ with noise $z \in p(z)$. Let what was called $f_w$ be the discriminator $D_w$. Then under suitable assumptions[36]

$$\nabla_\theta W(P_d, P_\theta) = -\nabla_\theta\, \mathbb{E}_{z\sim p(z)}[D_w(G_\theta(z))]$$
$$= -\mathbb{E}_{z\sim p(z)}[\nabla_\theta D_w(G_\theta(z))], \quad (8)$$

which is readibly computable. Here it is implicit that $D_w$ has been trained to be the function $f$ in (7) that appears in the approximation of $W(P_\theta, P_d)$. We then have a simple gradient for the generator update that approximates (if $D_w$ is perfectly trained) the Wasserstein gradient. This update, together with the discriminator update (7), form the basis of the Wasserstein GAN.[36]

We now see a significant qualitative departure from many GANs. In training typical GAN it is often possible to over-train the discriminator, leading to to poor gradient updates for the generator. For the generator gradient update in (8) for a WGAN, note instead the RHS only approximates the gradient of the Wasserstein distance (a useful gradient for training) when $D_w$ itself is well-trained. That is, the generator receives useful updates when the discriminator is strong. For this reason, the WGAN discriminator is often instead called a critic; its goal is not to be an adversary to the generator with which it competes, but instead an expert data connoisseur that helps the generator improve its behavior.

After this theoretical development, an intuitive understanding of the WGAN may be useful. Keeping in mind that the critic $D_w$ is trained to play the role of the function $f$ in (7), we see that it is training to obtain maximal separation between the real data and the fake data, i.e. maximizing the difference

$$\mathbb{E}_{x\sim P_d}[D_w(x)] - \mathbb{E}_{y\sim P_\theta}[D_w(y)] \in \mathbb{R}, \quad (9)$$

or alternatively minimizing its negative. The generator loss in (8) simply attempts to push $D_w(y) = D_w(G_\theta(z))$ the other direction, leading to a competition.

Finally, we introduce the conditional GAN[38] (cGAN). The idea behind the cGAN is rather simple: in some cases, one might like to simulate data with particular attributes. For instance, in the MNIST dataset of handwritten digits, a simple GAN can be utilized to generate fakes, but there is no control over which handwritten number is simulated. A cGAN solves the problem by passing a condition, such as the handwritten digit, through a (potentially trivial) parameterized function at input, whose output is concatenated with the usual noise $z \sim P(z)$ and

fed into another parameterized function. The combined function is the cGAN generator $G_\theta$, and the discriminator proceeds as usual.

In the cases that we study, the condition will be the $h^{11}$ of a Kähler metric that we wish to simulate. We will see that this allows for extrapolation to values of $h^{11}$ that are outside of the training set. We will be more precise about the encoding of the condition and the cGAN architecture when utilizing them in Section 3.2.

## 3. Random Tensor Approximations of String Effective Lagrangians

In this section we propose learning random tensor approximations (RTAs) of low energy Lagrangians that arise from string compactification and exemplify the idea for Kähler metrics on Kähler moduli space.

Let us first discuss why learning a RTA is relevant for approximate statistical predictions in string theory.

Computing observables $\mathcal{O}(i)$ associated with a string vacuum $i \in S_{\text{vac}}$ is facilitated by computing the Lagrangian $\mathcal{L}_i$ for low energy fluctuations around $i$. For instance, the $4d$ renormalizable Lagrangian for self interactions of canonically normalized scalar fluctuations $\phi^a$ around $i$ takes the form

$$\mathcal{L}_{s,i} = -\frac{1}{2}\partial_\mu\phi^a\partial^\mu\phi^b - M_{ab}\phi^a\phi^b - g_{abc}\phi^a\phi^b\phi^c - \lambda_{abcd}\phi^a\phi^b\phi^c\phi^d, \tag{10}$$

where the value of the coupling tensors $M$, $g$, and $\lambda$ are vacuum-dependent. In general, $\mathcal{L}_i$ also contains fields of other spins and associated coupling tensors. The couplings are critical in determining $\mathcal{O}(i)$, and therefore an essential element in making statistical predictions across $S_{\text{vac}}$ is having detailed knowledge, and ideally exact computations, for ensembles of coupling tensors.

However, the size of the landscape of vacua and its computational complexity together make constructing large ensembles of coupling tensors a laborious process. As a concrete example of the limitations, axion reheating was studied in a large ensemble of string compactifications with $N$ axion-like-particles (ALPs) in [10] and demonstrated to be asymmetric for all studied values of $N$. Computational limitations required restricting the exact calculations to $N \leq 200$, despite the fact that $N \sim O(2000)$ is generic in the known ensemble. One ulterior motive that we have for proposing the techniques in this work is to be able to estimate expectations for ALP-cosmology in the large $N$ regime.

Since directly computing large ensembles of coupling tensors is often intractable, it is natural to try to simulate them. This is what we mean by a random tensor approximation. In the language of machine learning, if $x \sim P_d(x)$ is a coupling tensor computed from some ensemble in string theory, one would like to learn a generative model $G_\theta$ such that noise samples $z \sim P(z)$ produce samples $G_\theta(z)$ of an implicit distribution $P_\theta$ that, after training, yields

$$P_\theta \simeq P_d. \tag{11}$$

For instance, if $P_d(x)$ is a distribution on the cubic couplings, one might draw noise $z$ from a multivariate Gaussian and train $G_\theta$ so that a batches of samples $z_i \sim P(z)$ yield simulated samples

$$\hat{g}_{i,abc} := G_\theta(z_i) \sim P_\theta, \tag{12}$$

such that $\hat{g}$ tensors are indistinguishable from $g$ tensors, according to some similarity measure.

We emphasize a crucial difference relative to previous applications of random matrix theory to the landscape: instead of hoping that a well-studied matrix ensemble approximates string data, we directly learn the random matrix ansatz using the data. This point will be discussed further in Section 4.

As a proof-of-principle, we wish to learn a RTA of an ensemble of couplings tensors arising in string theory.

Due to their intrinsic interest, we focus on Kähler metrics on the Kähler moduli space of Kreuzer-Skarke Calabi-Yau threefolds.[5] In this case, the tensors are matrices. As a first step, in this paper we will evaluate the Kähler metrics at the apex of the so-called stretched Kähler cone, as we wish to focus on learning the matrix ensemble across a diversity of topologies and extrapolating out of sample. In the future it would be interesting to attempt to learn the moduli dependence of the metric, which would amount to learning random matrix approximations to matrices of polynomial functions.

Let us first discuss the physics of Kähler metrics on Kähler moduli space. For concreteness we will choose to work in type IIB / F-theory. Consider a compactification of type IIB string theory on a Calabi-Yau threefold $X$. The Kähler moduli $T_i$ are the four-dimensional fields obtained by Kaluza-Klein reduction on $X$ as

$$T_i = \int_{D_i}\left(\frac{1}{2}J\wedge J + C_4\right) =: \tau_i + i\theta_i, \tag{13}$$

where $D_i$ are $h^{11}(X)$ divisors (four-cycles) that provide a basis for $H_4(X,\mathbb{Z})$ and $J = t_i\omega_i$ is the Kähler form, expressed in a basis $w_i \in H^{11}(X)$, and $C_4$ is the Ramond-Ramond four-form. The kinetic terms for the axion-like particles (ALPs) $\theta_i$ take the form

$$\mathcal{L}_{\theta,\text{kin}} = -M_p^2 K_{ij}\,\partial^\mu\theta^i\partial_\mu\theta^j, \tag{14}$$

and similarly for the saxions $\tau_i$. $K_{ij}$ is the metric on Kähler moduli space derived from the classical Kähler potential $\mathcal{K} = -2\log\mathcal{V}$, with derivatives taken with respect to $T_i$ and

$$\mathcal{V} = \int_X J\wedge J\wedge J = \frac{1}{6}\kappa^{ijk}t_it_jt_k \tag{15}$$

the overall volume of $X$ and $\kappa^{ijk}$ the triple intersection numbers on $X$. The Kähler metric is then $K_{ij} = \partial_i\partial_j\mathcal{K}$. The tree-level result receives quantum corrections due to worldsheet instantons, but the latter are negligible inside the stretched Kähler cone, which we will discuss momentarily.

From the structure of these equations, it is clear that the tree-level $K_{ij}$ is a matrix of polynomials in $t_i$. The polynomials themselves have detailed structure and properties derived from the topology of $X$ and its Kähler moduli space. As mentioned, we will content ourselves to evaluate $K_{ij}$ at points in the moduli space, that is for specific values of $t_i$. This does introduce a potential

source of sample bias into our studies (though for some applications it is not as severe as one might expect[10]). Our goal is to instead focus on diversity across different Calabi-Yau topologies rather than in the Kähler moduli space of a fixed-topology Calabi-Yau.

There is a simple way to see the physical importance of $K_{ij}$. Upon canonical normalization of the moduli, a change of basis in the fields makes the metric $\propto \delta_{ij}$, eigenvalues of $K_{ij}$ appear in all of the couplings in involving the new fields. The particle physics and cosmology implications of the ALPs therefore depend critically on $K_{ij}$. For instance, they can play a crucial role in asymmetric axion reheating[10] or couplings to the photon.[11]

Due to the technical fact that evaluating the inverse Kähler metric $K^{ij}$ is computationally easier, we will actually work with $K^{ij}$, instead of $K_{ij}$. In addition, in order to compare two Kähler metrics corresponding to two different geometries, we will normalize the metric such that the overall volume of $X$ is one. Extrapolating to other volumes is trivial, as the metric is a homogeneous function of the moduli.

With the above motivation and context in mind, for the remainder of the paper we will focus on learning random matrix approximations to $K^{ij}$.

To do so, we must have an ensemble to learn from. The algorithm we use to generate data is as follows:

**Algorithm 1** Generate ensemble of Kähler metrics

**Require:** Fixed value of $h^{11}$, set $S_{\text{poly}}$ of reflexive $4d$ polytopes with that value of $h^{11}$.

**for** polytope $P \in S_{\text{poly}}$ **do**
  FRST ← pushing triangulation of $P$.
  A ← ToricVariety(FRST).
  X ← generic anticanonical hypersurface in A.
  **if** $h^{11}(A) = h^{11}(X)$ **then**
    $J$ ← parameterized Kähler form.
    $\mathcal{V} \leftarrow \frac{1}{6}\int_X J \wedge J \wedge J$.
    $\mathcal{K} \leftarrow -2\log(\mathcal{V})$.
    $D_i$ ← toric divisor, where $i = 1, \ldots, h^{11} + 4$.
    $C_{ij} \leftarrow D_i \cdot D_j \cdot X, \quad \forall i, j$.
    $a$ ← point in Kähler moduli space such that $\text{vol}(\sum_{i,j} C_{ij})$ is minimized subject to the stretched Kähler cone condition $\text{vol}(C_{ij}) > 1\ \forall i, j$.
    $\tilde{a}$ ← rescale $a$ such that $\mathcal{V} = 1$.
    $K^{ij} \leftarrow (\partial_i \partial_j \mathcal{K})|_{\tilde{a}}^{-1}$.
    save $K^{ij}$ and its eigenvalues, which are $> 0$.
  **end if**
**end for**

The region in Kähler moduli space satisfying $\text{vol}(C_{ij}) \geq 1$ for all $i, j$ is the so-called stretched Kähler cone; the point $a$ is known as its apex. Both were introduced in [41].

For each $h^{11} \in \{10, 20, 22, 24, 26, 28, 30, 40, 50\}$ we studied the first 10,000 polytopes from [42]. Utilizing standard toric geometry packages in Sage, taking the fine regular star triangulation (FRST) gives an ambient space $A$, and the associated Calabi-Yau hypersurface $X$ is called favorable if $h^{11}(A) = h^{11}(X)$. Since we only study favorable cases, we simple refer to $h^{11} := h^{11}(X) = h^{11}(A)$. The number of favorable geometries is given in **Table 1**, split according to values of $h^{11}$ utilized in two different types of experiments. Further details for data generation can be found in the GitHub repository.[43]

**Table 1.** For each $h^{11}$, the number of favorable Calabi-Yau hypersurfaces associated to 10,000 toric ambient spaces obtained by pushing triangulations of $4d$ reflexive polytopes. Top and bottom are values of $h^{11}$ utilized in fixed $h^{11}$ and interpolation / extrapolation experiments, respectively.

| $h^{11}$ | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| # Favorable | 9282 | 5793 | 4222 | 5517 | 4899 |

| $h^{11}$ | 20 | 22 | 24 | 26 | 28 | 30 |
|---|---|---|---|---|---|---|
| # Favorable | 5793 | 4936 | 5152 | 3981 | 4074 | 1722 |

We will use this data in two different types of experiments, designed to test performance at fixed $h^{11}$, as well as the ability to interpolate or extrapolate out of sample, i.e. to values of $h^{11}$ not utilized in training.

We will also overcome a common problem with generative models based using the nature of Kähler metric data. The problem is that it is often unclear how to evaluate the performance of $G_\theta$. For instance, if $G_\theta$ is trained to provide deep fakes of human faces, the performance could be evaluated by asking humans to determine whether a set of samples if real or fake. However, this rather brittle process is expensive and slow. One would like a numerical figure-of-merit that may be easily computed and utilized to compare real data against fake data.

In our case, we will utilize the fact that our "images" are matrices that appear in low energy effective Lagrangians in string theory, the eigenvalue spectrum of which carries physical information. By contrast, it doesn't make sense to study the eigenvalue spectrum of human faces. Our figure-of-merit will be the distribution of $\log_{10}$ of the eigenvalues of $K^{ij}$, and specifically we will study how the Wasserstein distance between the real and fake log eigenspectra changes as the GANs are trained. That is, at fixed $h^{11}$ we produce $N_{\text{geom}}$ simulated inverse Kähler metrics $K^{ij}$, compute the log eigenspectrum of those samples, and compute the Wasserstein distance relative to the the test ensemble of real $K^{ij}$. (The test ensemble of Kähler metrics is the complement of the training ensemble inside the set of Kähler metrics on the favorable geometries in Table 3). One expects the eigenspectrum distance to decrease during training.

There is an obvious potential confusion that we would like clarify: this Wasserstein distance of log eigenspectra that is our figure-of-merit is completely separate from the Wasserstein distance implicit in WGANs. The latter is an approximate Wasserstein distance between $P_d$ and $P_\theta$, which is intractable due to the high dimension of the distributions and is therefore estimated using Kantorovich-Rubinstein duality. The former is simply the Wasserstein distance of the one-dimensional log eigenvalue distributions, and is readily computed using SciPy. *Specifically, we do not train on the log eigenvalue distribution.*

### 3.1. Kähler Metric Simulation at Fixed $h^{11}$

We first learn random matrix approximations of Kähler metrics at fixed values of $h^{11}$. The parameters available to our experiments are:

| Param. | Description |
|---|---|
| Model | GAN, WGAN, DCGAN, or DCWGAN |
| $h^{11}$ | Hodge number of $h^{11}(X)$ |
| $N_{\text{geom}}$ | # of geometries $X$ used in training |
| $n_z$ | # of draws from $\mathcal{N}(0, 1)$ at input |
| $N_{\text{batch}}$ | batch size |
| $N_{\text{crit}}$ | # of WGAN critic loops (if applicable) |
| $\alpha$ | learning rate for RMSProp |

where the GAN and WGAN in the model type denote the GAN loss and Wasserstein GAN loss introduced in Section 2. Both algorithms require a generator network and a discriminator network,

$$G_\theta : \mathbb{R}^{n_z} \to \mathbb{R}^{h^{11}} \times \mathbb{R}^{h^{11}}$$
$$D_w : \mathbb{R}^{h^{11}} \times \mathbb{R}^{h^{11}} \to D_T, \tag{16}$$

where in the Wasserstein GAN case the discriminator is often called the critic. $D_T$ is the discriminator target; for the GAN it is [0,1], and for the WGAN it is $\mathbb{R}$. The presence of DC in the model type denotes a deep convolutional architecture; otherwise it is a fully connected feed-forward network. Further details of the architecture can be found in the repository.[43]

In the case of a Wasserstein GAN, $N_{\text{crit}}$ is the number of batches the critic is trained on for each generator training batch. This parameter is crucial because, as discussed, the Wasserstein GAN requires a strong critic. If performance is poor, it may be due to a weak critic, which can be solved by increasing $N_{\text{crit}}$.

We run the first batch of experiments with fixed

$$(N_{\text{geom}}, N_{\text{batch}}, N_{\text{crit}}) = (2500, 64, 5), \tag{17}$$

models varying across the listed types, and

$$h^{11} \in \{10, 20, 30, 40, 50\},$$
$$\alpha \in \{5 \times 10^{-5}, 5 \times 10^{-6}\},$$
$$n_z \in \{5, 15, 25, 50\}, \tag{18}$$

for a total of 160 experiments, 1000 epochs each.

Results are presented in **Figure 1**, where we have focused on the $\alpha = 5 \times 10^{-6}$ since the lower learning rate decreases noise and clarifies the result. On top, we see that performance, as measured by the Wasserstein distance between the real and fake log eigenspectra, depends critically on the model type. A DCWGAN clearly performs best. This is not a surprise, as the Wasserstein GANs and / or deep convolutional architecture often improve GAN training. On the bottom, we see the the performance effectively does not depend on $n_z$ in the ranges we have chosen; note that performance does go down for $n_z = 1$, however. This point is worthy of significant discussion, see Section 4.

We also ran another experiment to aid in visualizing the results with respect to the actual images and the converging eigenspectra. The experiment is a DCWGAN with

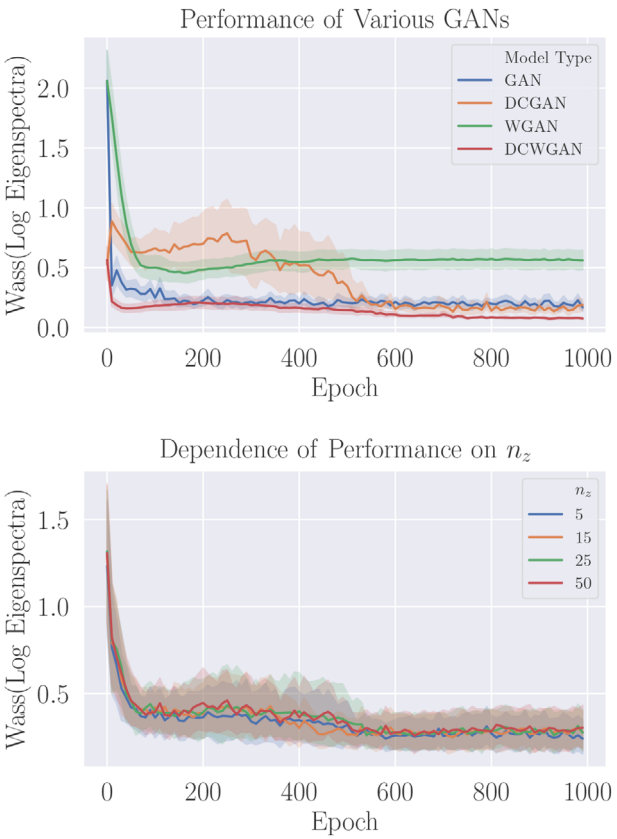


**Figure 1.** Performance of fixed $h^{11}$ experiments with $\alpha = 5 \times 10^{-6}$, dependent upon model type (left) and the number of Gaussian draws $n_z$ (right), with 95% confidence intervals. *Top:* a Wasserstein GAN with deep convolutional architecture gives the best performance and fastest training. *Bottom:* high accuracy simulation is achieved with little variance across the number of Gaussian draws, even with $n_z = 5 \ll h^{11} \in \{10, 20, 30, 40, 50\}$.

$$h^{11} = 10, \quad N_{\text{geom}} = 2500, \quad n_z = 5$$
$$N_{\text{batch}} = 64, \quad N_{\text{crit}} = 5, \quad \alpha = 2.5 \times 10^{-6}. \tag{19}$$

The progression of log eigenspectra and image representations during training are presented in **Figures 3** and **4**, respectively. In the former, the eigenspectra are seen to converge to good agreement. The plots also serve as a heuristic gauge for what Wasserstein distances of log eigenspectra correspond to good agreement between simulation and real data. To the naked eye, distances of $\lesssim .2$ have good agreement, whereas the distance .94 at epoch 0 demonstrates a poor model. In Figure 4, samples that were blurry and faint at early times increase in sharpness and contrast during training, looking increasingly realistic to the naked eye.

### 3.2. Interpolation and Extrapolation in $h^{11}$ with Conditional GANs

Since we would like to be able to reliably simulate string data in regimes where exact computation is intractable, we now study whether GANs for string data are able to interpolate and

extrapolate. Specifically, we study whether it is possible to interpolate or extrapolate in $h^{11}$, relative to the $h^{11}$ values of the training samples.

A priori this seems like a bad idea, because extrapolating out of sample is in general intractable, but in special cases it may be possible if the data is highly structured. This is often the case in string theory, and in the data that we study the structural relationship is due to topological transitions that change $h^{11}$. We will speculate about this further in Section 4.

Since we wish to interpolate and extrapolate, the techniques must differ in crucial ways from those of Section 3.1, though many of the parameters are the same.

First, we must introduce conditions, so that the input to the generative model is not only noise $z \sim P(z)$, but also some information about the nature of the sample we wish to generate. For us, it is $h^{11}$ that we wish to pass as a condition. We one-hot encode[5] the value of $h^{11}$ and pass it through a function:

$$C_\phi : \mathbb{Z}^k \to \mathbb{R}^l \tag{20}$$

where $l$ is a hyperparameter and $C$ may have non-linearities. The noise input $z \sim P(z)$ is concatenated with $C_\phi(c)$ for $c \in \mathbb{Z}^k$ and passed as input to

$$N_\varphi : \mathbb{R}^{l+n_z} \to \mathbb{R}^{\max h^{11}} \times \mathbb{R}^{\max h^{11}}, \tag{21}$$

which together form the generator

$$G_\theta : \mathbb{Z}^k \times \mathbb{R}^{n_z} \to \mathbb{R}^{\max h^{11}} \times \mathbb{R}^{\max h^{11}} \tag{22}$$

via $G_\theta(c, z) = N_\varphi(C_\phi(c), z)$, so that the parameters $\theta$ are the union of $\varphi$ and $\phi$. For us, $C_\phi$ is a fully-connected layer with LeakyReLU activation and $N_\varphi$ is effectively one of the $G_\theta$ of Section 3.1, together with some additional zero-padding since the data is not uniform, due to varying $h^{11}$. For architecture details, see the repository.[43]

Second, we must state the relationship between interpolation, extrapolation, and the conditions. If the values of $h^{11}$ utilized during training and testing are

$$\begin{aligned} h^{11}_{\text{train}} &= \{20, 22, 28, 30\} \\ h^{11}_{\text{test}} &= \{20, 22, 24, 26, 28, 30\}, \end{aligned} \tag{23}$$

then the set $h^{11}_{\text{test}} \setminus h^{11}_{\text{train}} = \{24, 26\}$ means that we test also for $h^{11}$ values that are in between the training values; this is interpolation. Similarly, if

$$\begin{aligned} h^{11}_{\text{train}} &= \{20, 22, 24, 26\} \\ h^{11}_{\text{test}} &= \{20, 22, 24, 26, 28, 30\}, \end{aligned} \tag{24}$$

then accurate predictions for $h^{11}_{\text{test}} \setminus h^{11}_{\text{train}} = \{28, 30\}$ corresponds to extrapolation. Again our figure-of-merit is the Wasserstein distance of the log eigenspectra of $K_{ij}$, but now there are six comparisons, one for each $h^{11} \in h^{11}_{\text{test}}$, two of which do not appear in the train set. We are testing if the cGAN simulate Kähler metrics for values of $h^{11}$ not involved in training.

Given the success of the Wasserstein DCGAN in simulating Kähler metrics at fixed $h^{11}$, we promote this model alone to become a conditional GAN, so that the full model we study for interpolation and extrapolation is a conditional deep convolutional Wasserstein GAN. This means that $N_\varphi$ is a deep convolutional network and the associated generator $G_\theta$ and $D_w$ are trained as a Wasserstein GAN. The parameters are

| Param. | Description |
|---|---|
| $h^{11}_{\text{train}}$ | $h^{11}$ values of training set |
| $h^{11}_{\text{test}}$ | $h^{11}$ values of test set |
| $N_{\text{geom}}$ | # geometries $X$ used in training per $h^{11}$ |
| $n_z$ | # of draws from $\mathcal{N}(0, 1)$ at input |
| $N_{\text{batch}}$ | batch size |
| $N_{\text{crit}}$ | # of WGAN critic loops (if applicable) |
| $\alpha$ | learning rate for RMSProp |
| $l$ | width of latent layer encoding for $h^{11}$ |

In our experiments, $h^{11}_{\text{train}}$ and $h^{11}_{\text{test}}$ as chosen as in (23) and (24) for interpolation and extrapolation, and we take $k = \max h^{11} = \max(h^{11}_{\text{test}})$. Furthermore we take

$$(N_{\text{geom}}, N_{\text{batch}}, N_{\text{crit}}, \alpha) = (2500, 64, 5, 10^{-7}), \tag{25}$$

and

$$n_z \in \{10, 25, 50, 100\}, \tag{26}$$

for a total of 4 different experiment types for interpolation, and 4 for extrapolation. We found that these experiments, perhaps due to the complexity of the input and architecture, lead to more noise, and we therefore ran each of these experiments 10 times to build statistics.

Results are presented in **Figure 2**. From the mean performance plots, we see clear evidence of learning across all values of $h^{11} \in h^{11}_{\text{test}}$, with the trend that learning is a bit modest for small $h^{11}$. The decreasing performance with decreasing $h^{11}$ is likely due to the fact that the smaller the value of $h^{11}$, the more zero-padding is necessary. For instance, a metric with $h^{11} = 20$ has 400 entries, but since $30 \in h^{11}_{\text{test}}$ it is zero-padded such that it has 500 more zeroes than every metric with $h^{11} = 30$. This effect is almost certainly solvable with a more clever architecture that allows for non-uniform data. Nevertheless, learning occurs for all values of $h^{11}$.

Two specific experiments are also presented, to demonstrate trends that are common in many of the experiments. Specifically, experiments that start with a large Wasserstein loss often have significant learning in the first $O(200)$ epochs, but then experience a bump that decreases the performance, particularly at smaller $h^{11}$. This is sometimes overcome with additional learning at late times that leads to the best results, as demonstrated on the RHS of Figure 2. In some cases the experiments start with relatively low Wasserstein loss, in which case significant learning does not necessarily occur.

Most notably, as is the point of this section, we emphasize that these generative models demonstrate the ability to

[5] A one-hot encoding of an integer $i$ represents $i$ by the unit vector $e_i \in \mathbb{Z}^k$, where there are $k$ different allowed values of $i$.

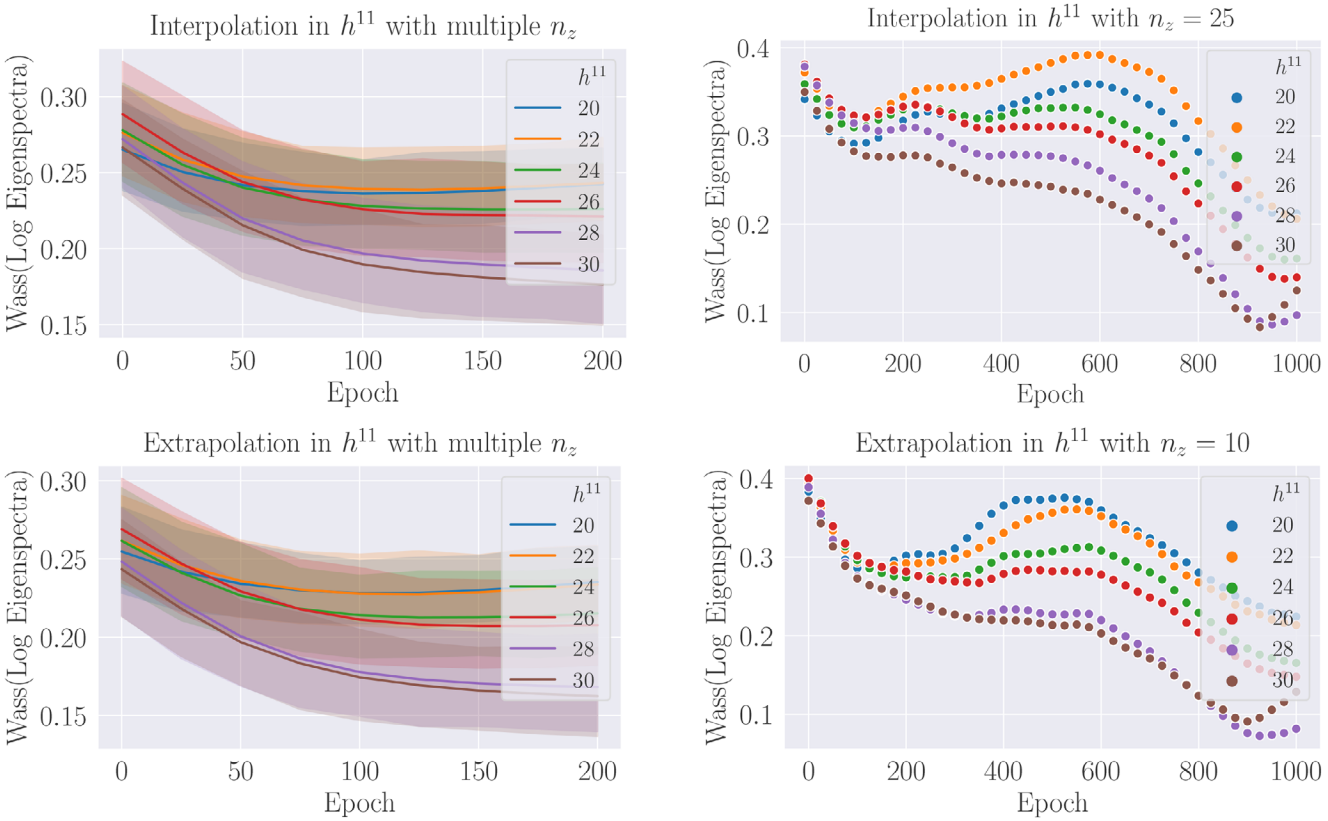


**Figure 2.** Performance of interpolation (top) and and extrapolation (bottom) experiments, as a function of the parameter $h^{11}$ supplied as a condition to the GAN. *Left:* Mean performance and 95% confidence intervals. *Right:* Illustrative single experiments with stated $n_z$ and learning rates of $10^{-7}$.

interpolate and extrapolate to metrics at values of $h^{11}$ that were not involved in training. Specifically, in the top two plots of Figure 2 the learning associated with the $h^{11} \in \{24, 26\}$ data demonstrates the ability of the cGAN to interpolate, while the bottom two plots exhibit extrapolation due to the learning associated with the $h^{11} \in \{28, 30\}$ data. Reasons that we did not push the technique further will be addressed in Section 4, but we consider this a successful proof-of-principle of the ability of generative models to exhibit some extrapolation on string theory data, perhaps due to structural topological relationships between geometries.

Our GAN approach allows for fast simulation. The trained conditional DCWGAN provides a speedup of generation of Kähler metrics at $h^{11} = 30$ by a factor of about 250, compared to the current leading pipeline[6] from polytope to effective Lagrangian,[44] and so yields a large speedup at fixed $h^{11}$. Importantly, while the pipeline in [44] will have at least a polynomial-time slowdown with $h^{11}$, the speed of the conditional DCWGAN is fixed across all $h^{11}$, since it is input to a fixed trained neural network. Therefore, if one can actually use this technique to extrapolate to large $h^{11}$, the conditional DCWGAN will likely provide a means to sample effective Lagrangians at large $h^{11}$ where no other technique will be fast enough to provide useful statistics.

[6] We thank Mehmet Demirtas for performing a computation of Kähler metrics to which we can compare our results.

## 4. Discussion

In this paper we have introduced a new approach to making statistical predictions in string theory. We proposed the use of deep generative models, a class of techniques in machine learning that train a generator function (deep neural network) $G_\theta$ to convert draws from a distribution $P(z)$ to draws from a distribution $P_\theta$ that approximates a data distribution $P_d$. Specifically, we utilized generative adversarial networks (GANs), but this is simply an instantiation of the broader idea, and it is worth exploring other possibilities as well.

To see the utility of such techniques in string theory, consider what one would do in the presence of an all-powerful oracle with perfect knowledge of the string landscape. The oracle knows the full set of vacua $S_{\text{vac}}$ and the cosmological probability distribution $P(i)$ on it. It can efficiently sample $P(i)$ and compute any observable $O(i)$ for any $i \in S_{\text{vac}}$. Then there is no obstacle to making statistical predictions: one simply uses the oracle to collect enough samples from $P(i)$, computes ensemble averages of observables, and compares to experiment.

Of course, this oracle is rather futuristic. We currently only know subsets of $S_{\text{vac}}$, albeit very large ones, and despite some progress there is still much to be understood about dynamical and anthropic contributions to $P(i)$. Furthermore, in some classes of vacua it is not known how to compute some basic observables, or it is simply inefficient, sometimes due to running up against

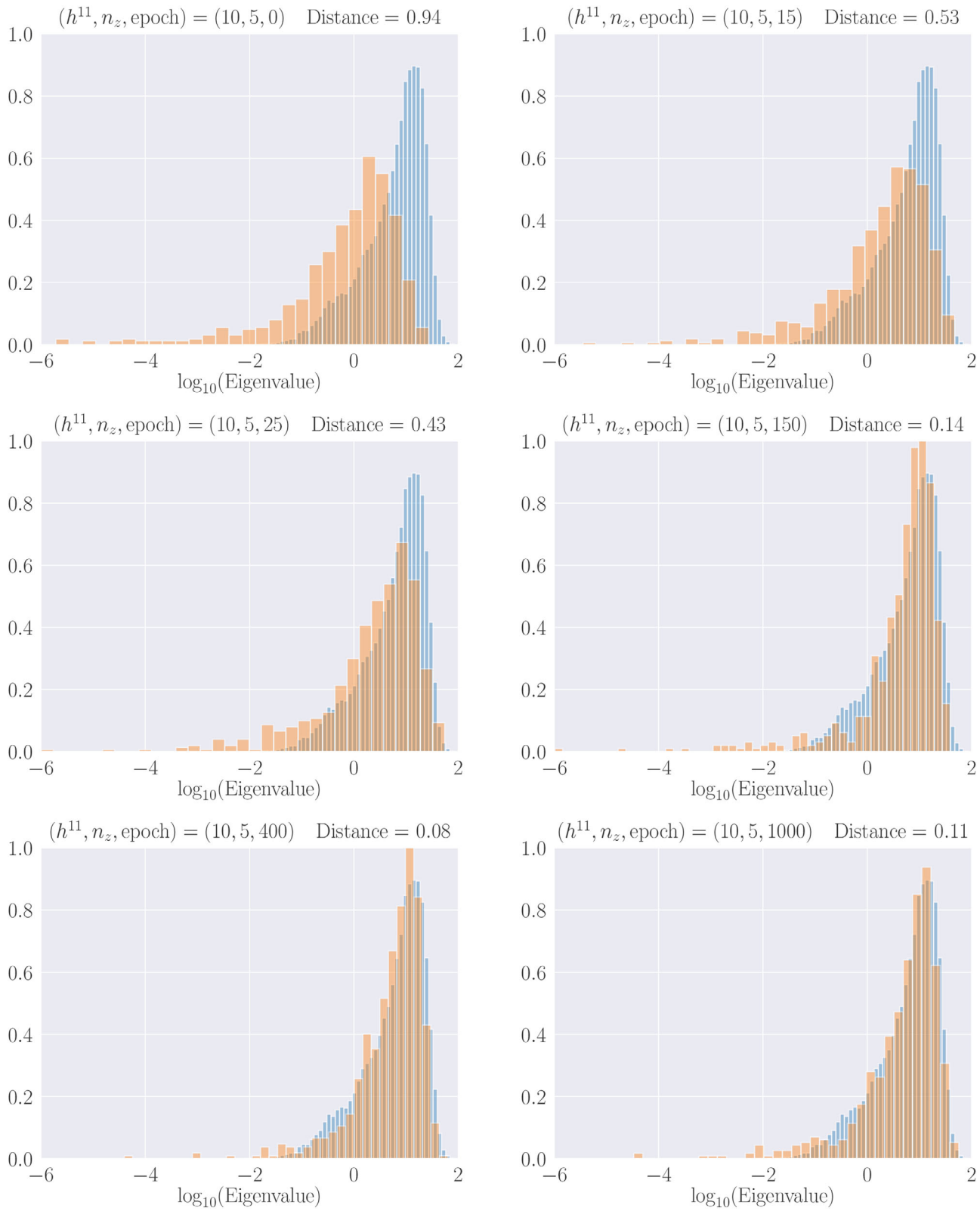


**Figure 3.** Eigenspectrum change under training an $n_z = 5$ Wasserstein DCGAN with $h^{11} = 10$. Blue is the ground truth Kähler metric eigenspectrum from Calabi-Yau compactification. Orange is the eigenspectrum of the simulated Kähler metrics. Note the overshoot in the peak while approaching epoch 400, but its subsequent flattening as training continues to epoch 1000.

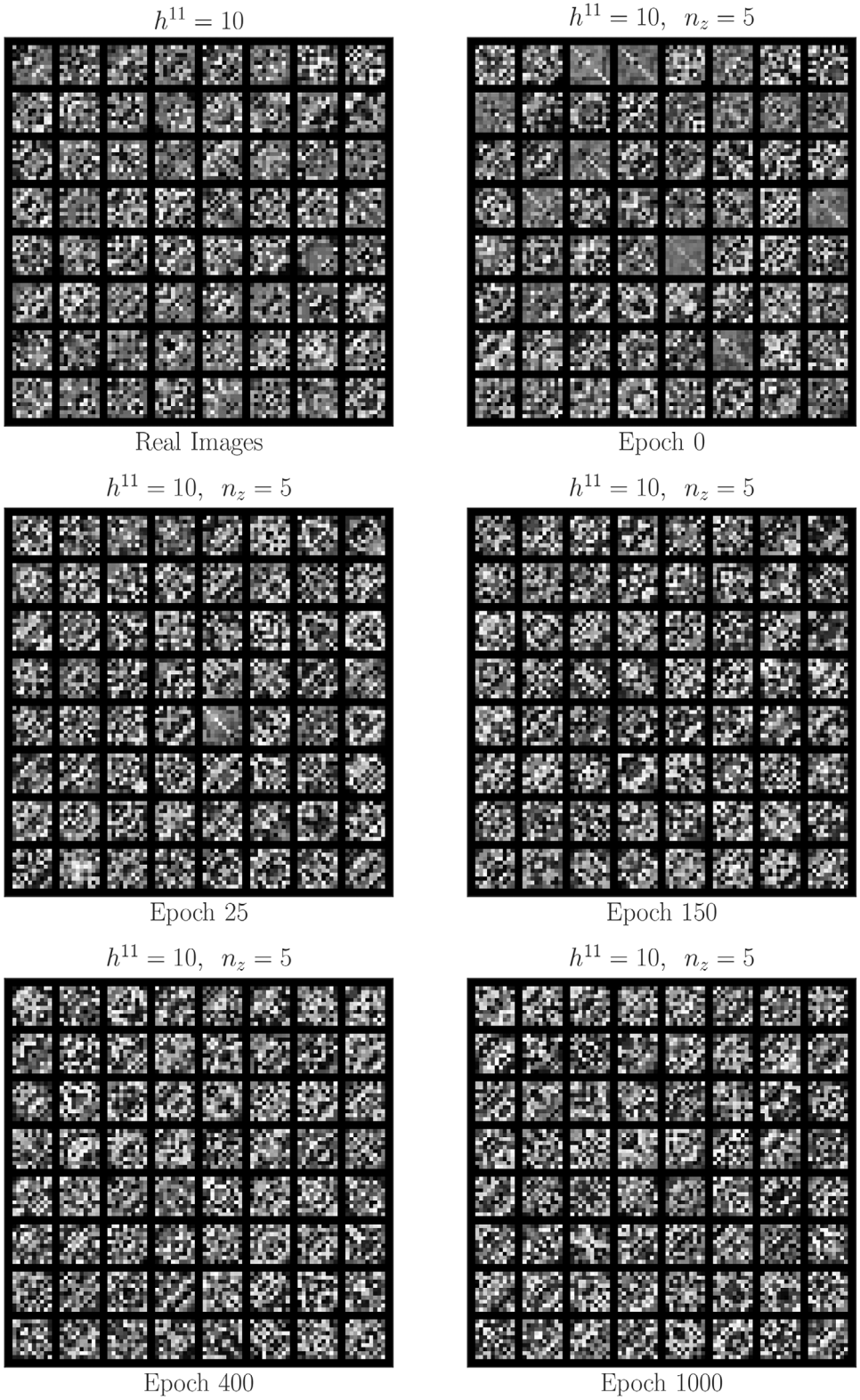


**Figure 4.** Image representation with fixed noise inputs under training an $n_z = 5$ Wasserstein DCGAN with $h^{11} = 10$, with ground truth Kähler metrics in the upper left and the rest simulation. Each graphic presents 64 Kähler metrics, each a $10 \times 10$ image. Samples that are faint and blurry at early times become increasingly sharp and realistic during training.

instances of NP-hard problems. Even a weaker oracle that only knows $S_{\rm vac}$, $P(i)$, and how to compute observables has a serious problem: sampling is non-trivial, yet crucial to making statistical predictions.

By learning a distribution $P_\theta$ that approximates a data distribution $P_d$ and generating samples from $P_\theta$, generative models offer the possibility of trading some error for efficient sampling. If the error is sufficiently small and/or controllable, this provides a useful means for making *approximate* statistical predictions in string theory. That is the central conceptual idea in this paper.

There is a down-to-earth application of this idea that we explored. For vacua whose low energy fluctuations admit a Lagrangian description, learning to approximately sample them corresponds to learning a random tensor approximation (RTA) for the couplings in the Lagrangian. For two-index couplings, this is simply learning a random matrix approximation (RMA).

This is markedly different from previous applications of random matrix theory (RMT) in or inspired by the string landscape.[45–49] There, it was often the case that well-studied random matrix ensembles were studied at large $N$ (number of fields, cycles, etc) and universality yielded physical implications. However, it is not clear a priori why such ensembles should have anything to do with string theory, which exhibits structures that may violate assumptions of certain RMT ensembles. In fact, observables in known ensembles of the type IIB theory compactified on Calabi-Yau manifolds deviate[47] from the expectations of canonical RMT ensembles.

Instead, generative models offer a means of *learning* RTAs of string effective Lagrangians.

We exemplified the idea in the case of Kähler metrics on the Kähler moduli space of Calabi-Yau manifolds. Such metrics were generated for thousands of Kreuzer-Skarke Calabi-Yau threefolds at various values of $h^{11}$, which served as training data from which to learn random tensor approximations. We utilized multiple different types of GANs, which differ according to their loss functions (a normal GAN versus a WGAN) and architecture (fully connected feedforward versus convolutional). In each case, the GAN generator is a deep neural network $G_\theta$ that produces simulated Kähler metrics as $G_\theta(z)$, where $z$ is a vector of noise of dimension $n_z$ with entries drawn from the Gaussian distribution $\mathcal{N}(0,1)$.

Unlike in many applications of GANs, we have a natural figure-of-merit by which to judge the learning process: the Wasserstein distance between the log eigenvalue distributions of the real and fake Kähler metrics. Given $N$ real Kähler metrics arising from Calabi-Yau manifolds and $N$ fake Kähler metrics $G_\theta(z)$, with $N$ sufficiently large, we compute the log eigenspectrum. A bad RMA of Kähler metrics will have significant mismatch between the log eigenspectra. If learning is occurring as $G_\theta$ is trained then they should increasingly overlap, which we measure with the Wasserstein (a.k.a. earth-mover) distance; as discussed, this use of Wasserstein distance is fundamentally different from that of the WGAN.

We performed two classes of experiments that demonstrate learning of RMAs of Kähler metrics.

**Fixed $h^{11}$ results:** In Section 3.1, we trained GANs at fixed values of $h^{11} \in \{10, 20, 30, 40, 50\}$. In all cases, the Wasserstein distance on log eigenspectra decreases significantly during training. We found that a Wasserstein GAN with deep convolutional architecture (DCWGAN) significantly outperforms the other GAN types that we tried. When viewing the Kähler metrics as grayscale images, we found that at early times some of the images were faint with low contrast relative to the real Kähler metrics. This improved upon training; i.e., some aspects of learning can be seen with the naked eye.

Perhaps the most important result for the fixed $h^{11}$ experiments is that taking different values of $n_z \in \{5, 15, 25, 50\}$ did little to affect performance, at least with respect to the Wasserstein distance on the log eigenspectra. This is rather remarkable: despite the disparate $h^{11}$ values and thousands of geometries utilized for each, the neural network is able to generate matrices whose eigenspectrum resembles the Calabi-Yau data using only 5 Gaussian draws,[7] where $5 \ll h^{11}$. This suggests that the so-called "data manifold" is of relatively small dimension, demonstrating implicit correlations.

It is worth commenting further on this data manifold in light of the difference between RMAs and the well-studied random matrix ensembles previously applied in the string literature. For instance, it might be considered natural to use the Wishart ensemble to model Kähler metrics, since its matrices are also positive definite.[8] However, due to the $N^2$ i.i.d. entries the Wishart ensemble has dimension $N^2$ support in the space of $N \times N$ matrices. This is clearly different from Kähler metrics on Kähler moduli space (with $N = h^{11}$), which despite being $N \times N$ matrices nevertheless only depend on $h^{11}$ variables: the Kähler moduli. On general grounds, then, one should not expect the Wishart ensemble to be a good approximation to Kähler metrics.

In this light, we revisit the fact that even $n_z = 5$ GANs yielded good simulations of Kähler metrics. From the fact that they are functions of Kähler moduli space, one expects that $n_z = h^{11}$ draws should suffice, given a sufficiently expressive neural network, but in fact $n_z \ll h^{11}$ seems to do rather well. Clearly this cannot be exactly true, since the exact (rather than approximated) ensemble of tree-level Kähler metrics depends explicitly on a manifold of dimension $h^{11}$, the Kähler moduli space. This deserves further thought, and we will return to it after summarizing another result.

**Extrapolation in $h^{11}$ results:** In the second class of experiments, studied in Section 3.2, we studied whether the GAN had the ability to interpolate or extrapolate out of sample. This is of interest because computational complexity often limits exact computations to moderate $N$ regimes (see, e.g., the ALP example in the text), despite the fact most vacua are expected to live at large $N$. Clearly it would be beneficial if a GAN could simulate string data at large $N$, if exact computations are not available there. While a priori one should be skeptical of such extrapolation, it may perhaps be possible if the data is highly structured, as it often is in string theory.

To attempt interpolation and extrapolation, we used a conditional GAN (cGAN) with Wasserstein loss function and deep convolutional architecture; a cDCWGAN, putting the pieces together. The key difference in a conditional GAN is that the input is not simply noise drawn from some distribution, but also

[7] Note that one can try to take this too far: performance goes down significantly for $n_z = 1$, for instance.

[8] A Wishart matrix is of the form $A^T A$, where the matrix $A$ has its entries independently and identically distributed (i.i.d.) according to a Gaussian distribution.

a condition that dictates what *type* of sample to generate. For instance, in generating handwritten digits, one might wish to have the ability to choose whether to generate a seven or a nine. For us, we passed $h^{11}$ as a condition, so that the GAN learns to simulate Kähler metrics at a chosen value of $h^{11}$. Interpolation (extrapolation) then corresponds to the accurate generation of Kähler metrics (as measured by Wasserstein distance of log eigenspectra) for values of $h^{11}$ in between (larger than) the values of $h^{11}$ of the real Kähler metrics used in training. Specifically, in the interpolation experiments we trained at $h^{11} \in \{20, 22, 28, 30\}$ and tested for those values, as well as the interpolated values $h^{11} \in \{24, 26\}$. For extrapolation, we trained at $h^{11} \in \{20, 22, 24, 26\}$ and tested at those values, and also the extrapolated values $h^{11} \in \{28, 30\}$.

The result is that the GAN learned to generate Kähler metrics at values of $h^{11}$ not utilized in training, i.e. it was able to both interpolate and extrapolate. There was decreased performance for smaller $h^{11}$, almost certainly correlated with increased amounts of zero-padding for smaller $h^{11}$; this can likely be overcome by utilizing architectures that allow for non-uniform data. We also point out that we did not attempt to extrapolate further in $h^{11}$, since as $h^{11}$ increases the eigenvalue distribution for Kähler metrics becomes bimodal, and thus far we have found it difficult to model the second mode, though we expect this is doable with future advances. It would also be interesting to understand geometric origin of the second mode, which may be related to qualitative changes (such as increasing numbers of facet interior points) of the associated reflexive polytopes as $h^{11}$ increases.

**Concluding comments**. Following our proposal for making approximate statistical predictions in string theory, we have presented concrete results that demonstrate the ability of generative models to simulate string data. Though we specifically used GANs to simulate Kähler metrics, there is no clear obstruction preventing the use of other generative models or studying other types of data, including structures in formal theory that may not be as relevant for the landscape. For instance, one could utilize normalizing flows, which not only give the ability to generate samples, but also allow for the computation of the probability of the sample in $P_\theta$, due to the generative model being an invertible neural network.

The neural networks performed surprisingly well in at least two ways. First, with very few random draws, $n_z = 5 \ll h^{11}$, they were able to efficiently simulate Kähler metrics at fixed $h^{11}$. Second, the conditional GAN was able to extrapolate, simulating Kähler metrics at values of $h^{11}$ not seen during training. From a machine learning perspective, these facts suggest the presence of structure that is making learning possible.

Perhaps it is the highly structured and relational nature of string data the makes it learnable. Not only is a single data point typically accompanied by significant structure, such as the topological and geometric information carried by a fixed string vacuum, but these data points are related to one another by deformations or discrete operations in a mathematically rigorous space, such as moduli spaces relating fixed string geometries as well as topological transitions between them.

To that end, we would like to end with a speculation. In algebraic geometry there is a conjecture, known as Reid's fantasy, that all Calabi-Yau manifolds (of fixed dimension) are continuously connected by metric deformations and topology changing transitions. Many expect Reid's fantasy to be true, and if so there is a structural relationship between all Calabi-Yau manifolds that relates them to one another. In that case, it is reasonable to speculate that machine learning techniques might implicitly utilize the structural relationships to achieve better-than-expected learning. Perhaps we are seeing the first evidence of it with the results presented in this work.

## Acknowledgements

We thank Ana Achúcarro, Kyle Cranmer, Mehmet Demirtas, Mohamed El Amine Seddik, Tej Kanwar, Sven Krippendorf, Andre Lukas, Liam McAllister, Fabian Ruehle, Gary Shiu, Alexander Westphal, and especially Danilo Rezende for discussions regarding this work. Portions of this work were completed at the Aspen Center for Physics, which is supported by National Science Foundation grant PHY-1607611. J.H. and C.L. are supported by NSF CAREER grant PHY-1848089.

## Conflict of Interest

The authors have declared no conflict of interest.

## Keywords

deep learning, string theory

Received: January 10, 2020
Published online: April 18, 2020

[1] R. Bousso, J. Polchinski, *JHEP* **2000**, *06*, 006, arXiv:hep-th/0004134 [hep-th].
[2] F. Denef, M. R. Douglas, *JHEP* **2004**, *05*, 072, arXiv:hep-th/0404116 [hep-th].
[3] F. Denef, M. R. Douglas, *JHEP* **2005**, *03*, 061, arXiv:hep-th/0411183 [hep-th].
[4] W. Taylor, Y.-N. Wang, *JHEP* **2015**, *12*, 164, arXiv:1511.03209 [hep-th].
[5] M. Kreuzer, H. Skarke, *Adv. Theor. Math. Phys.* **2002**, *4*, 1209, arXiv:hep-th/0002240 [hep-th].
[6] J. Halverson, C. Long, B. Sung, *Phys. Rev.* **2017**, *D96*, 126006, arXiv:1706.02299 [hep-th].
[7] a) W. Taylor, Y.-N. Wang, *JHEP* **2018**, *01*, 111, arXiv:1710.11235 [hep-th]; b) R. Altman, J. Carifio, J. Halverson, B. D. Nelson, *JHEP* **2019**, *03*, 186, arXiv:1811.06490 [hep-th].
[8] M. R. Douglas, *JHEP* **2003**, *05*, 046, arXiv:hep-th/0303194 [hep-th].
[9] J. Halverson, B. D. Nelson, F. Ruehle, *Phys. Rev.* **2017**, *D95*, 043527, arXiv:1609.02151 [hep-ph].
[10] J. Halverson, C. Long, B. Nelson, G. Salinas, *Phys. Rev.* **2019**, *D99*, 086014, arXiv:1903.04495 [hep-th].
[11] J. Halverson, C. Long, B. Nelson, G. Salinas, *Phys. Rev.* **2019**, *D100*, 106010, arXiv:1909.05257 [hep-th].
[12] J. Garriga, D. Schwartz-Perlov, A. Vilenkin, S. Winitzki, *JCAP* **2006**, *0601*, 017, arXiv:hep-th/0509184 [hep-th].
[13] a) A. De Simone, A. H. Guth, A. D. Linde, M. Noorbala, M. P. Salem, A. Vilenkin, *Phys. Rev.* **2010**, *D82*, 063520, arXiv:0808.3778 [hep-th]; b) J. Carifio, W. J. Cunningham, J. Halverson, D. Krioukov, C. Long, B. D. Nelson, *Phys. Rev. Lett.* **2018**, *121*, 101602, arXiv:1711.06685 [hep-th].
[14] R. Bousso, B. Freivogel, I.-S. Yang, *Phys. Rev.* **2006**, *D74*, 103516, arXiv:hep-th/0606114 [hep-th].
[15] R. Bousso, B. Freivogel, *JHEP* **2007**, *06*, 018, arXiv:hep-th/0610132 [hep-th].

 

[16] R. Bousso, B. Freivogel, I.-S. Yang, *Phys. Rev.* **2009**, *D79*, 063513, arXiv:0808.3770 [hep-th].
[17] B. Freivogel, *Class. Quant. Grav.* **2011**, *28*, 204007, arXiv:1105.0244 [hep-th].
[18] F. Denef, M. R. Douglas, B. Greene, C. Zukowski, *Annals Phys.* **2018**, *392*, 93, arXiv:1706.06430 [hep-th].
[19] J. Khoury, O. Parrikar, **2019**, arXiv:1907.07693 [hep-th].
[20] J. Khoury, **2019**, arXiv:1912.06706 [hep-th].
[21] S. Weinberg, *Phys. Rev. Lett.* **1987**, *59*, 2607.
[22] F. Denef, M. R. Douglas, *Annals Phys.* **2007**, *322*, 1096, arXiv:hep-th/0602072 [hep-th].
[23] M. Cvetic, I. Garcia-Etxebarria, J. Halverson, *Fortsch. Phys.* **2011**, *59*, 243, arXiv:1009.5386 [hep-th].
[24] J. Halverson, F. Ruehle, *Phys. Rev.* **2019**, *D99*, 046015, arXiv:1809.08279 [hep-th].
[25] J. Halverson, M. Plesser, F. Ruehle, J. Tian, **2019**, arXiv:1911.07835 [hep-th].
[26] Y.-H. He, **2017**, arXiv:1706.02714 [hep-th].
[27] D. Krefl, R.-K. Seong, *Phys. Rev.* **2017**, *D96*, 066014, arXiv:1706.03346 [hep-th].
[28] F. Ruehle, *JHEP* **2017**, *08*, 038, arXiv:1706.07024 [hep-th].
[29] J. Carifio, J. Halverson, D. Krioukov, B. D. Nelson, *JHEP* **2017**, *09*, 157, arXiv:1707.00655 [hep-th].
[30] a) J. Liu, *JHEP* **2017**, *12*, 149, arXiv:1707.02800 [hep-th]; b) K. Hashimoto, S. Sugishita, A. Tanaka, A. Tomiya, *Phys. Rev.* **2018**, *D98*, 046019, arXiv:1802.08313 [hep-th]; c) Y.-N. Wang, Z. Zhang, *JHEP* **2018**, *08*, 009, arXiv:1804.07296 [hep-th]; d) R. Jinno, **2018**, arXiv:1805.12153 [hep-th]; e) K. Bull, Y.-H. He, V. Jejjala, C. Mishra, *Phys. Lett.* **2018**, *B785*, 65, arXiv:1806.03121 [hep-th]; f) A. Constantin, A. Lukas, *Fortsch. Phys.* **2019**, *67*, 1900084, arXiv:1808.09992 [hep-th]; g) D. Klaewer, L. Schlechter, *Phys. Lett.* **2019**, *B789*, 438, arXiv:1809.02547 [hep-th]; h) T. Rudelius, *JCAP* **2019**, *1902*, 044, arXiv:1810.05159 [hep-th]; i) A. Mütter, E. Parr, P. K. S. Vaudrevange, *Nucl. Phys.* **2019**, *B940*, 113, arXiv:1811.05993 [hep-th]; j) Y.-H. He, **2018**, arXiv:1812.02893 [hep-th]; k) A. Cole, G. Shiu, *JHEP* **2019**, *03*, 054, arXiv:1812.06960 [hep-th]; l) V. Jejjala, A. Kar, O. Parrikar, **2019**, https://doi.org/10.1016/j.physletb.2019.135033, arXiv:1902.05547 [hep-th]; m) K. Bull, Y.-H. He, V. Jejjala, C. Mishra, *Phys. Lett.* **2019**, *B795*, 700, arXiv:1903.03113 [hep-th]; n) K. Hashimoto, *Phys. Rev.* **2019**, *D99*, 106017, arXiv:1903.04951 [hep-th]; o) J. Halverson, B. Nelson, F. Ruehle, *JHEP* **2019**, *06*, 003, arXiv:1903.11616 [hep-th]; p) Y.-H. He, S.-J. Lee, *Phys. Lett.* **2019**, *B798*, 134889, arXiv:1904.08530 [hep-th]; q) Y.-H. He, M. Kim, **2019**, arXiv:1905.02263 [cs.LG]; r) A. Cole, A. Schachner, G. Shiu, *JHEP* **2019**, *11*, 045, arXiv:1907.10072 [hep-th]; s) A. Ashmore, Y.-H. He, B. A. Ovrut, **2019**, arXiv:1910.08605 [hep-th]; t) E. Parr, P. K. S. Vaudrevange, **2019**, arXiv:1910.13473 [hep-th]; u) L. Alessandretti, A. Baronchelli, Y.-H. He, **2019**, arXiv:1911.02008 [math.NT].
[31] F. Ruehle, *Physics Reports* **2019**, https://doi.org/10.1016/j.physrep.2019.09.005.
[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, in *Advances in Neural Information Processing Systems 27* (Eds: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger), Curran Associates, Inc., **2014**, pp. 2672–2680.
[33] H. Erbin, S. Krippendorf, **2018**, arXiv:1809.02612 [cs.LG].
[34] D. P. Kingma, M. Welling, "Auto-encoding variational bayes," **2013**, arXiv:1312.6114 [stat.ML].
[35] D. J. Rezende, S. Mohamed, "Variational inference with normalizing flows," **2015**, arXiv:1505.05770 [stat.ML].
[36] M. Arjovsky, S. Chintala, L. Bottou, arXiv e-prints, arXiv:1701.07875 **2017**, arXiv:1701.07875 [stat.ML].
[37] A. Radford, L. Metz, S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks,"**2015**, arXiv:1511.06434 [cs.LG].
[38] M. Mirza, S. Osindero, *CoRR* **2014**, *abs/1411.1784*, arXiv:1411.1784.
[39] M. Lucic, K. Kurach, M. Michalski, S. Gelly, O. Bousquet, "Are gans created equal? a large-scale study,"**2017**, arXiv:1711.10337 [stat.ML].
[40] C. Villani, "Optimal transport – old and new," **2008**, pp. xxii+973.
[41] M. Demirtas, C. Long, L. McAllister, M. Stillman, **2018**, arXiv:1808.01282 [hep-th].
[42] "Kreuzer-skarke database," http://hep.itp.tuwien.ac.at/kreuzer/CY/, accessed: 2019–12-30.
[43] "GANs for Kähler metrics," https://github.com/jimhalverson/gans_for_kahler_metrics, to appear early 2020.
[44] M. Demirtas, L. McAllister, A. Rios Tascon, (A Triangulation Survey at Large Hodge Numbers, to appear).
[45] D. Marsh, L. McAllister, T. Wrase, *JHEP* **2012**, *03*, 102, arXiv:1112.3034 [hep-th].
[46] a) X. Chen, G. Shiu, Y. Sumitomo, S. H. H. Tye, *JHEP* **2012**, *04*, 026, arXiv:1112.3338 [hep-th]; b) F. G. Pedro, A. Westphal, *Phys. Lett.* **2014**, *B739*, 439, arXiv:1303.3224 [hep-th].
[47] a) C. Long, L. McAllister, P. McGuirk, *JHEP* **2014**, *10*, 187, arXiv:1407.0709 [hep-th]; b) A. Achúcarro, P. Ortiz, K. Sousa, *Phys. Rev.* **2016**, *D94*, 086012, arXiv:1510.01273 [hep-th]; c) F. G. Pedro, A. Westphal, *Phys. Rev.* **2017**, *E95*, 032144, arXiv:1606.07768 [cond-mat.stat-mech]; d) F. G. Pedro, A. Westphal, *JHEP* **2017**, *03*, 163, arXiv:1611.07059 [hep-th].
[48] T. C. Bachlechner, D. Marsh, L. McAllister, T. Wrase, *JHEP* **2013**, *01*, 136, arXiv:1207.2763 [hep-th].
[49] T. C. Bachlechner, *JHEP* **2014**, *04*, 054, arXiv:1401.6187 [hep-th].

 