
Learning Size and Shape of Calabi-Yau Spaces

Magdalena Larfors

Department of Physics and Astronomy
Uppsala University
SE-751 20 Uppsala
Sweden
magdalena.larfors@physics.uu.se

Andre Lukas

Rudolf Peierls Centre for Theoretical Physics
Oxford University
Parks Road, Oxford OX1 3PU
United Kingdom
andre.lukas@physics.ox.ac.uk

Fabian Ruehle

Department of Physics
Northeastern University
360 Huntington Avenue, Boston MA 02115
United States
f.ruehle@northeastern.edu

Robin Schneider

Department of Physics and Astronomy
Uppsala University
SE-751 20 Uppsala
Sweden
robin.schneider@physics.uu.se

Abstract

We present a new machine learning library for computing metrics of string compactification spaces. We benchmark the performance on Monte-Carlo sampled integrals against previous numerical approximations and find that our neural networks are more sample- and computation-efficient. We are the first to provide the possibility to compute these metrics for arbitrary, user-specified shape *and* size parameters of the compact space and observe a linear relation between optimization of the partial differential equation we are training against and vanishing Ricci curvature.

1 Introduction

String theory is the leading candidate for a unification of general relativity and the standard model of particle physics. Its mathematical consistency, however, requires the introduction of six additional small and compact dimensions, usually described by Calabi-Yau (CY) manifolds. The shape, size, and topology of these internal spaces determine many interesting physical quantities such as the gauge group and particle content of the theory. CY manifolds are not only interesting objects in physics, but exhibit fascinating mathematical properties, such as mirror symmetry.

CY spaces have been extensively studied, starting with [1, 2]. However, above four compact dimensions, there exists no known closed form expression for their metric tensor. The metric is of utmost importance to connect string theory with observable physics. It is a necessary ingredient in the computation of physical Yukawa couplings from string theory and can be used to test conjectures from the swampland program [3, 4] or regarding mirror symmetry [5]. These applications furthermore require control over *how* the metric depends on the size and shape parameters, so-called *moduli*, of the CY geometry.

The standard numerical approximation for CY metrics is the Donaldson algorithm [6], a fixed point iteration methods which expands the metric in a monomial basis of degree k and converges to the CY metric as $k \rightarrow \infty$. This algorithm has been utilized in examples [7–12]. However, there are three main drawbacks: first, its time and space complexity scales poorly (factorially) with k . Second, changing the moduli of the metric requires running the whole fixed point iteration from scratch. Lastly, the algorithm is in fact providing a metric with *different* properties, which happens to coincide with the CY metric in the limit $k \rightarrow \infty$. However, better direct optimizations of the CY metric are possible for any finite, fixed k . Energy functionals [13, 14] and holomorphic neural networks [15] have proven more efficient. Alternatively, one may directly learn the CY metric tensor [16–18]. By modeling the metric as neural network, this has also allowed studies of the shape moduli dependence [17].

In this paper we advance this promising line of research by presenting an open source package, `cymetric`¹, to learn CY metrics with TensorFlow [19] on a wide class of CY manifolds. Given the lack of analytic metric solutions, direct supervised training of the neural network is impossible. We therefore proceed in an almost self-supervised manner and introduce five loss functions that govern the learning process. Our neural network reproduces the full moduli dependence of the CY metric, thus filling a crucial knowledge gap which has so far impeded phenomenological studies in string theory. The implementation improves sample and computation efficiency compared to previous approximations. Moreover, experiments run with our package establish a direct linear relation between optimizing a simple surrogate loss and the computationally much more expensive vanishing Ricci curvature.

The results of our experiments have promising implications for utilizing neural networks in solving differential equations [20, 21]. In particular, we demonstrate that by using physical domain knowledge, it is possible to find solutions to non-linear second-order differential (Monge-Ampère) equations by computing only a single derivative in the training process. There are (besides computation time) no limitations on generating points samples as input data, and each sample point comes with machine precision. This opens up the possibility to study various scaling laws of neural networks [22]. The training data consists of Monte-Carlo sampled points on the CY. Using a powerful theorem of Shiffman and Zelditch [23], we do not need to perform MCMC, but can construct the measure directly by embedding the manifold into a higher-dimensional ambient space with known measure.

2 Methodology

Calabi-Yau manifolds A CY manifold, X , is a compact, complex, Ricci-flat Kähler manifold. To describe it, we recall a few items from differential, complex and Kähler geometry. A CY geometry is specified by two nowhere-vanishing differential forms: a real Kähler 2-form J , associated to a Kähler metric g , and a holomorphic n -form Ω , which specifies how n holomorphic coordinates are selected among the $2n$ real ones. Given X , the Kähler form is not unique; however, within the same cohomology class of any closed J' , there is unique representative J whose corresponding metric g has vanishing Ricci curvature by the celebrated Calabi-Yau theorem [24, 25]. On any Kähler manifold, the metric is encoded in a Kähler potential \mathcal{K} , and the Ricci curvature simplifies to

$$g = \partial\bar{\partial}\mathcal{K}, \quad R = \partial\bar{\partial}\log \det g. \quad (1)$$

The unique volume form of a CY three-fold can be expressed in terms of either J or Ω . Combining this with the cohomological equivalence of J and J' , we get the *Monge-Ampère equation*,

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

for a real scalar function ϕ , where κ is some complex constant. This second order complex non-linear partial differential equation is satisfied iff g is the unique Ricci-flat metric on X [24, 25].

¹<https://github.com/pythoncymetric/cymetric>

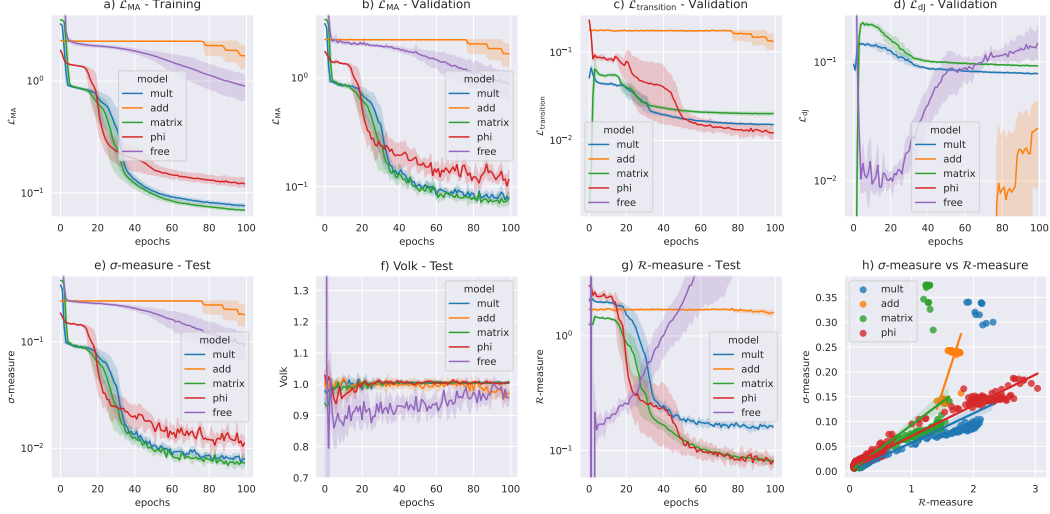


Figure 1: Fermat Quintic experiments: a) Monge-Ampère loss on training data; b)+c)+d) Monge-Ampère, transition and Kähler loss on validation data; e) σ -measure f) volume and g) \mathcal{R} -measure on test data; h) the linear relationship between improvement in σ -measure and \mathcal{R} -measure.

CY manifolds are commonly constructed as (intersections of) hypersurfaces, i.e. the vanishing loci of polynomial equations, in a higher-dimensional ambient space, \mathcal{A} . In this paper we consider complete intersection CY manifolds in $\mathcal{A} = \prod_{i=1}^r \mathbb{P}^{n_i}$, where \mathbb{P}^{n_i} are complex projective spaces [1, 2]. Our TensorFlow models also work for the numerous CY hypersurfaces in toric varieties specified in the Kreuzer-Skarke list [26] and for CY manifolds in complex dimension other than three. The performance of our model is comparable for projective and toric ambient spaces, but is reduced for CY manifolds that are either defined by multiple hypersurfaces, or have many shape moduli.

Point Sampling The training data consists of points on the CY X , which must be sampled uniformly with respect to a known measure. Rejection sampling is not feasible due to the high dimensionality of our problem. MCMC methods can deal with high dimensionality [27], but they require rejecting many points, and so are not optimal. We therefore implemented a more efficient point sampling method in NumPy [28], which employs theorems from complex geometry [8, 23], and applies to CY spaces constructed from intersecting hypersurfaces. This results in points sampled according to a measure dA , which is computed as an anti-symmetric product of certain ambient space measures pulled back to the CY. A function f may then be Monte-Carlo integrated over the CY using

$$\int_X d\text{Vol}_{\text{CY}} f = \int_X \frac{d\text{Vol}_{\text{CY}}}{dA} dA f = \frac{1}{N} \sum_i w_i f|_{p_i} \quad \text{with} \quad w_i = \frac{d\text{Vol}_{\text{CY}}}{dA} |_{p_i}, \quad (3)$$

where the measure dA and weights w_i are given by the outlined procedure. Finally, the performance of numeric metric approximations is evaluated by integrating equations (1) and (2) over X using (3).

Neural Networks The metric tensor g is a $(3, 3)$ hermitian complex tensor. The obvious ansatz is to let a neural network (NN) freely predict nine real numbers, $g_{\text{pr}} = g_{\text{NN}}$, such that we have a mapping $f : \mathbb{R}^{2 \sum n_i + 1} \rightarrow \mathbb{R}^9$. As we will show in the next section, this is not the most efficient way to approach this problem, and one should modify the prediction of the NN with physical intuition. Equation (2) shows that the unique Ricci-flat metric on X is given by an exact correction to some reference Kähler metric g_{FS} , which we may construct by pullback from the ambient space $\mathcal{A} = \prod_{i=1}^r \mathbb{P}^{n_i}$.² We then define NNs that learn corrections to g_{FS} such that the resulting metric g_{pr} is Ricci-flat: addition $g_{\text{pr}} = g_{\text{FS}} + g_{\text{NN}}$, element-wise multiplication $g_{\text{pr}} = g_{\text{FS}} + g_{\text{FS}} \odot g_{\text{NN}}$, matrix product $g_{\text{pr}} = g_{\text{FS}} + g_{\text{FS}} \cdot g_{\text{NN}}$, and Monge-Ampère type $g_{\text{pr}} = g_{\text{FS}} + \partial\bar{\partial}\phi$.

² \mathcal{A} admits a metric, \tilde{g}_{FS} , known as a Fubini-Study metric. The pullback $g_{\text{FS}} = I_a^i(\tilde{g}_{\text{FS}})_{i\bar{j}} I_b^{\bar{j}}$ (where $I_a^i = \partial z_i / \partial x_a$, and z_i and x_a are coordinates on \mathcal{A} and X) yields a globally well-defined Kähler metric on X .

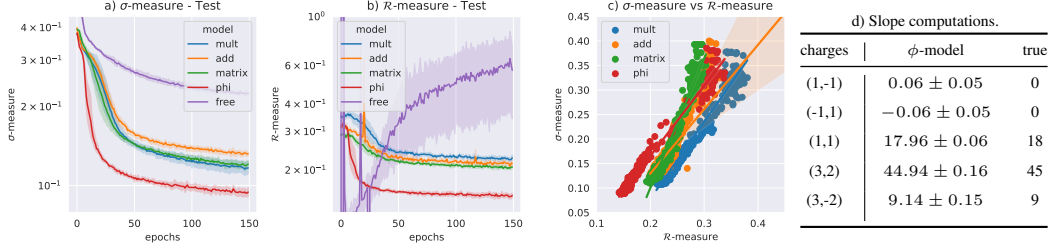


Figure 2: Bicubic experiments: a) σ -measure b) \mathcal{R} -measure and c) the linear relationship between improvement in σ -measure and \mathcal{R} -measure and d) mean slope (with one standard deviation as error) computations for five line bundles on test data.

Loss contributions We define five loss functions that ensure that the numerical metric satisfies the Monge-Ampère equation (2), is Kähler, i.e. $dJ = 0$, is well defined over patch transitions, has vanishing Ricci curvature (1), and results in the overall volume as the reference metric g_{FS} (this is a necessary condition to preserve the Kähler class). The total training loss is thus

$$\mathcal{L} = \alpha_1 \mathcal{L}_{MA} + \alpha_2 \mathcal{L}_{dJ} + \alpha_3 \mathcal{L}_{\text{transition}} + \alpha_4 \mathcal{L}_{\text{Ricci}} + \alpha_5 \mathcal{L}_{\text{vol-K}} \quad (4)$$

where α_i are hyperparameters (by default set to $\alpha_i = 1.0$), and

$$\mathcal{L}_{MA} = \left\| \left| 1 - \frac{1}{\kappa} \frac{\det g_{pr}}{\Omega \wedge \bar{\Omega}} \right| \right\|_n, \quad \mathcal{L}_{\text{Ricci}} = \|R\|_n = \|\partial\bar{\partial} \ln \det g_{pr}\|_n, \quad (5)$$

$$\mathcal{L}_{dJ} = \sum_{ijk} \|\text{Re } c_{ijk}\|_n + \|\text{Im } c_{ijk}\|_n, \quad \text{with } c_{ijk} = g_{i\bar{j},k} - g_{k\bar{j},i} \quad \text{and } g_{i\bar{j},k} = \partial_k g_{i\bar{j}} \quad (6)$$

$$\mathcal{L}_{\text{transition}} = \frac{1}{d} \sum_{(s,t)} \left\| g_{pr}^{(t)} - T_{(s,t)} \cdot g_{pr}^{(s)} \cdot T_{(s,t)}^\dagger \right\|_n \quad \text{and } \mathcal{L}_{\text{vol-K}} = \left\| \int \det \tilde{g}_{FS} - \int \det g_{pr} \right\|_n. \quad (7)$$

Here the subscript n denote the L_n norms (default is $n = 1$ for all but \mathcal{L}_{dJ} which has $n = 2$), $T_{(s,t)} = \partial \bar{x}^{(s)} / \partial \bar{x}^{(t)}$ denote transition matrices between patches s and t , d is the number of patch transitions, and the integral is taken over mini-batches. By virtue of the Calabi-Yau theorem, \mathcal{L}_{MA} is a surrogate loss (which does not involve costly derivatives) for the Ricci-loss $\mathcal{L}_{\text{Ricci}}$; hence the latter is disabled by default. Moreover, the ϕ -model is by construction Kähler with fixed size moduli, and thus comes with a disabled \mathcal{L}_{dJ} and $\mathcal{L}_{\text{vol-K}}$. It does requires two derivatives with respect to the input parameters instead, which makes it comparable in speed to the other NNs. The derivatives with respect to the input coordinates are computed with TensorFlow's automatic differentiation, which works reliable even when training the ϕ -model against the Ricci-scalar with a total of five nested gradient tapes.

3 Experiments

Fermat Quintic To illustrate our package and benchmark it against previous results we first consider the Fermat Quintic, which is defined as the zero locus of $Q = \sum_{i=0}^4 z_i^5$ in $\mathcal{A} = \mathbb{P}^4$. Figure 1 shows the results for each of the five model ansätze introduced in the previous section averaged over five runs. Training is done with an Adam optimizer on a fully connected feed-forward network with three hidden layers of 64 units each and GELU activation function for 100 epochs. Since we only want to learn a correction to the known Fubini-Study metric, we initialize the NN weights with small Gaussian variance $\mathcal{N}(0, 0.01)$ ($\mathcal{N}(0, 1)$ for the free network). The training set contains 198.000 points with a 0.1 validation split. We performed hyperparameter tuning via box searches for the number of hidden layers and activation function.

As seen in Figure 1, the two multiplication networks and the ϕ -model consistently arrive at the Ricci-flat metric. We evaluate the performance of the NN on the established benchmarks [10, 11] of sigma and Ricci measures, (σ, \mathcal{R}) , with an additional test set of 22.000 points in subplots e) and g). The ϕ -models with lowest validation loss reach, within a few hours on a laptop, a mean accuracy of $\sigma = 0.0086$ and $\mathcal{R} = 0.076$. This is on par with $k = 20$ in Donaldson algorithm [11] with a training

time of 35 years on 4.6E8 points [16]. We are the first to systematically study \mathcal{R} , and demonstrate, in subplot h), a linear relation, $\sigma \approx 0.06\mathcal{R}$, between optimization of the surrogate Monge-Ampère equation (2) and decrease in Ricci measure \mathcal{R} . Finally, subplot f) shows that the normalized volume of the manifold is constant, and hence that we do not change the Kähler class during training.

Bicubic As a second example we consider the bicubic manifold X given by a homogeneous degree (3,3) polynomial in $\mathcal{A} = \mathbb{P}^2 \times \mathbb{P}^2$ and size moduli, $t_1 = 1 = t_2$. For phenomenological reasons, and also to improve performance, we choose a member of this family which is invariant under a \mathbb{Z}_3 symmetry in its shape moduli. Figure 2 shows the averaged results of five experiments using the previously described experimental setup. As for the quintic, the three “multiplicative” NNs that are learning corrections relative to the value of the reference metric g_{FS} outperform the free and additive NNs, whose output is not scaled with the reference metric and varies over several orders of magnitude. The ϕ -model is particularly successful and the linear relation persists, $\sigma \approx 1.4\mathcal{R} - 0.1$.

For CYs with multiple size moduli t_i , an important consistency check of the t_i -dependence of the NN is given by the so-called slope, which is defined as $\mu = \int \sqrt{g} g^{a\bar{b}} F_{a\bar{b}}$ for a vector bundle V with curvature F . Clearly, this slope depends not only on the CY volume, encoded in \sqrt{g} , but on the full metric tensor that is approximated by the NN. Crucially, μ may also be computed exactly using topological data of (X, V) , and can in fact be shown to vanish for stable, holomorphic V [29, 30]. Subtable d) of Figure 2 shows that the slope computed using the ϕ -model for various line bundles is in excellent agreement with the exact, topological results, both for stable and non-stable bundles.

4 Conclusion

We have presented a computational library that learns Ricci-flat metrics for a wide range of CY manifolds. The package comprises five conceptually different NN architectures, and we compare their efficacy in learning the metric. Two novel features compared to previous studies, are that the numerical metrics can be computed for any value of the CY shape and size moduli, and the package allows for direct studies of the Ricci scalar. Our package has a significantly lower entry barrier than previous methods. Moreover, using domain knowledge in point sampling and during training greatly improves the performance. We emphasize that our results can be further improved by more careful hyperparameter optimization, more sample points, or simply longer training time.

Broader Impact

We have more efficient algorithms than before, which reduces the carbon footprint of numerical computations for phenomenological and mathematical studies of string theory. We democratize the learning of Calabi-Yau metrics by creating an open source package with low entry barrier, to the benefit of researchers that have smaller computational budgets. This may in part offset the positive environmental effect just mentioned, as energy consumption will grow with the number of users of the package. The package can be used to test swampland conjectures which are actively discussed in the string theory community. This might open up resources for other applications down the line.

Acknowledgments and Disclosure of Funding

RS and ML are funded in part by the Swedish Research Council (VR) under grant number 2020-03230. The work of FR is supported by startup funding from Northeastern University. Computations were in part enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at the HPC cluster *Tetralith*, partially funded by the Swedish Research Council through grant agreement no. 2018-05973, the hydra cluster of the Physics Department at the University of Oxford, and the computing cluster at CERN.

References

- [1] P. Candelas, A. M. Dale, C. A. Lutken, and R. Schimmrigk. Complete Intersection Calabi-Yau Manifolds. *Nucl. Phys.*, B298:493, 1988. doi:10.1016/0550-3213(88)90352-5.

- [2] Paul S. Green, Tristan Hubsch, and Carsten A. Lutken. All Hodge Numbers of All Complete Intersection Calabi-Yau Manifolds. *Class. Quant. Grav.*, 6:105–124, 1989. doi:10.1088/0264-9381/6/2/006.
- [3] Cumrun Vafa. The String landscape and the swampland. 9 2005. URL: <https://arxiv.org/abs/hep-th/0509212>, arXiv:hep-th/0509212.
- [4] Anthony Ashmore and Fabian Ruehle. Moduli-dependent KK towers and the swampland distance conjecture on the quintic Calabi-Yau manifold. *Phys. Rev. D*, 103(10):106028, 2021. arXiv:2103.07472, doi:10.1103/PhysRevD.103.106028.
- [5] Andrew Strominger, Shing-Tung Yau, and Eric Zaslow. Mirror symmetry is T duality. *Nucl. Phys. B*, 479:243–259, 1996. arXiv:hep-th/9606040, doi:10.1016/0550-3213(96)00434-8.
- [6] S. K. Donaldson. Some numerical results in complex differential geometry. *arXiv Mathematics e-prints*, page math/0512625, December 2005. arXiv:math/0512625.
- [7] Michael R. Douglas, Robert L. Karp, Sergio Lukic, and Rene Reinbacher. Numerical Calabi-Yau metrics. *J. Math. Phys.*, 49:032302, 2008. arXiv:hep-th/0612075, doi:10.1063/1.2888403.
- [8] Volker Braun, Tamaz Brelidze, Michael R. Douglas, and Burt A. Ovrut. Calabi-Yau Metrics for Quotients and Complete Intersections. *JHEP*, 05:080, 2008. arXiv:0712.3563, doi:10.1088/1126-6708/2008/05/080.
- [9] Volker Braun, Tamaz Brelidze, Michael R. Douglas, and Burt A. Ovrut. Eigenvalues and Eigenfunctions of the Scalar Laplace Operator on Calabi-Yau Manifolds. *JHEP*, 07:120, 2008. arXiv:0805.3689, doi:10.1088/1126-6708/2008/07/120.
- [10] Michael R. Douglas, Robert L. Karp, Sergio Lukic, and Rene Reinbacher. Numerical solution to the hermitian Yang-Mills equation on the Fermat quintic. *JHEP*, 12:083, 2007. arXiv:hep-th/0606261, doi:10.1088/1126-6708/2007/12/083.
- [11] Lara B. Anderson, Volker Braun, Robert L. Karp, and Burt A. Ovrut. Numerical Hermitian Yang-Mills Connections and Vector Bundle Stability in Heterotic Theories. *JHEP*, 06:107, 2010. arXiv:1004.4399, doi:10.1007/JHEP06(2010)107.
- [12] Lara B. Anderson, Volker Braun, and Burt A. Ovrut. Numerical Hermitian Yang-Mills Connections and Kahler Cone Substructure. *JHEP*, 01:014, 2012. arXiv:1103.3041, doi:10.1007/JHEP01(2012)014.
- [13] Matthew Headrick and Ali Nassar. Energy functionals for Calabi-Yau metrics. *Adv. Theor. Math. Phys.*, 17(5):867–902, 2013. arXiv:0908.2635, doi:10.4310/ATMP.2013.v17.n5.a1.
- [14] Wei Cui and James Gray. Numerical Metrics, Curvature Expansions and Calabi-Yau Manifolds. *JHEP*, 05:044, 2020. arXiv:1912.11068, doi:10.1007/JHEP05(2020)044.
- [15] Michael R. Douglas, Subramanian Lakshminarasimhan, and Yidi Qi. Numerical Calabi-Yau metrics from holomorphic networks. 12 2020. URL: <https://arxiv.org/abs/2012.04797>, arXiv:2012.04797.
- [16] Anthony Ashmore, Yang-Hui He, and Burt A. Ovrut. Machine Learning Calabi-Yau Metrics. *Fortsch. Phys.*, 68(9):2000068, 2020. arXiv:1910.08605, doi:10.1002/prop.202000068.
- [17] Lara B. Anderson, Mathis Gerdes, James Gray, Sven Krippendorff, Nikhil Raghuram, and Fabian Ruehle. Moduli-dependent Calabi-Yau and SU(3)-structure metrics from Machine Learning. *JHEP*, 05:013, 2021. arXiv:2012.04656, doi:10.1007/JHEP05(2021)013.
- [18] Vishnu Jejjala, Damian Kaloni Mayorga Pena, and Challenger Mishra. Neural Network Approximations for Calabi-Yau Metrics. 12 2020. URL: <https://arxiv.org/abs/2012.15821>, arXiv:2012.15821.

- [19] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. URL: <https://www.tensorflow.org/>.
- [20] Henry Jin, Marios Mattheakis, and Pavlos Protopapas. Unsupervised neural networks for quantum eigenvalue problems. *NeurIPS Machine Learning for Physical Sciences*, 2020. URL: <https://arxiv.org/abs/2010.05075>, arXiv:2010.05075.
- [21] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [22] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. URL: <https://arxiv.org/abs/2001.08361>, arXiv:2001.08361.
- [23] Bernard Shiffman and Steve Zelditch. Distribution of zeros of random and quantum chaotic sections of positive line bundles. *Communications in Mathematical Physics*, 200(3):661–683, January 1999. arXiv:math/9803052, doi:10.1007/s002200050544.
- [24] Eugenio Calabi. *On Kähler Manifolds with Vanishing Canonical Class*, pages 78–89. Princeton University Press, 2015. URL: <https://doi.org/10.1515/9781400879915-006>, doi:doi:10.1515/9781400879915-006.
- [25] Shing-Tung Yau. On the ricci curvature of a compact kähler manifold and the complex monge-ampère equation, I. *Commun. Pure Appl. Math.*, 31(3):339–411, 1978. doi:10.1002/cpa.3160310304.
- [26] Maximilian Kreuzer and Harald Skarke. Complete classification of reflexive polyhedra in four-dimensions. *Adv. Theor. Math. Phys.*, 4:1209–1230, 2002. arXiv:hep-th/0002240, doi:10.4310/ATMP.2000.v4.n6.a2.
- [27] Matthew D Hoffman, Andrew Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.
- [28] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi:10.1038/s41586-020-2649-2.
- [29] Simon K Donaldson. Anti self-dual yang-mills connections over complex algebraic surfaces and stable vector bundles. *Proceedings of the London Mathematical Society*, 3(1):1–26, 1985.
- [30] Karen Uhlenbeck and Shing-Tung Yau. On the existence of hermitian-yang-mills connections in stable vector bundles. *Communications on Pure and Applied Mathematics*, 39(S1), 1986.

Checklist

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [\[Yes\]](#)

- (b) Did you describe the limitations of your work? [Yes]
 - (c) Did you discuss any potential negative societal impacts of your work? [Yes]
 - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
- (a) Did you state the full set of assumptions of all theoretical results? [N/A] Not applicable.
 - (b) Did you include complete proofs of all theoretical results? [N/A] Not applicable.
3. If you ran experiments...
- (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The package can be found here: <https://github.com/pythoncymetric/cymetric>
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
- (a) If your work uses existing assets, did you cite the creators? [Yes]
 - (b) Did you mention the license of the assets? [No] We will use GPL-3 for the open source package.
 - (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Yes, see link to source code above.
 - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Not applicable.
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Not applicable.
5. If you used crowd sourcing or conducted research with human subjects...
- (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Not applicable.
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Not applicable.
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Not applicable.