Identifying equivalent Calabi–Yau topologies: A discrete challenge from math and physics for machine learning

Vishnu Jejjala∗1, Washington Taylor†2, and Andrew P. Turner‡3

1Mandelstam Institute for Theoretical Physics, School of Physics, NITheCS, and CoE-MaSS, University of the Witwatersrand, 1 Jan Smuts Avenue, Johannesburg, WITS 2050, South Africa

2Center for Theoretical Physics, Department of Physics; NSF AI Institute for Artificial Intelligence and Fundamental Interactions, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

3Department of Physics and Astronomy, University of Pennsylvania, 209 South 33rd Street, Philadelphia, PA 19104, USA

February 16, 2022

Abstract: We review briefly the characteristic topological data of Calabi–Yau threefolds and focus on the question of when two threefolds are equivalent through related topological data. This provides an interesting test case for machine learning methodology in discrete mathematics problems motivated by physics.
1 Calabi–Yau threefolds

Calabi–Yau threefolds are a class of geometric spaces that have been studied intensively for several decades by mathematicians and physicists in the context of superstring compactification [1, 2, 3]. The key features of these real 6-dimensional manifolds are (1) that they admit a Ricci-flat Kähler metric, and thus solve the vacuum Einstein equations $R_{\mu\nu} = 0$, and (2) they are Kähler manifolds and thus have a complex structure compatible with supersymmetry. This makes them useful as geometries for compactifying superstring theory from ten dimensions to the physical four dimensions of space-time. Mathematically, the key feature of a Calabi–Yau variety is the vanishing of the canonical class $K = 0$ (up to torsion).

Despite extensive investigations on these geometries, a fundamental open question remains: Is the number of topologically distinct Calabi–Yau threefolds finite or infinite? [4]

Topologically, a theorem of Wall [5] states that a Calabi–Yau threefold $X$ is uniquely determined by:

- $h^{1,1}, h^{2,1} \in \mathbb{Z}_+$: Hodge numbers;
- $C_{ijk} \in \mathbb{Z}$, $i, j, k \in \{1, \ldots, h^{1,1}\}$: triple intersection numbers between divisors (complex codimension-one subvarieties) in the linear space $H^{1,1}(X, \mathbb{Z})$ (where $h^{1,1}(X) = \dim H^{1,1}(X, \mathbb{Z})$);
- $p_1(T_X) = 2c_2(T_X)$: The first Pontryagin class, a vector of $h^{1,1}$ integers.

Thus, the topological data of a Calabi–Yau threefold is characterized by a finite list of integer data. Fundamental questions to which the answers are not yet known are:

1. Which sets of integer data are allowed?
2. Which sets of integer data are equivalent under a basis change for $H^{1,1}$ (and hence describe the same geometry)?

We are focused here on the second of these questions.

Over the years, many large classes of Calabi–Yau threefolds have been constructed. The largest is the set of toric hypersurface Calabi–Yau threefolds realized through the Batyrev construction [6] as hypersurfaces in toric fourfolds; Kreuzer and Skarke have compiled a comprehensive list of 473, 800, 776 reflexive 4D polytopes that provide such constructions [7, 8]. Other constructions include complete intersection Calabi–Yau (CICY) threefolds [9] and generalized CICYs [10].

Regarding the finiteness question, recent work on elliptic Calabi–Yau threefolds, which are particularly relevant in the context of F-theory [11, 12], has given some fairly strong evidence that the number of distinct topological Calabi–Yau threefolds may be finite (up
Figure 1: Hodge numbers of the 473.8 million families of Calabi–Yau threefolds described by Kreuzer and Skarke through toric hypersurface constructions from 4D reflexive polytopes. Hodge numbers for the less than 30,000 cases without obvious elliptic fibers are shown in red; even many of these may have more subtle elliptic fibration structure in some phase. Since the number of topologically distinct elliptic Calabi–Yau threefolds is finite, this suggests finiteness of the full Calabi–Yau landscape.

An elliptic (or more generally genus one fibered) Calabi–Yau threefold admits a fibration $\pi: X \to B_2$ so that over an open subset of $B_2$ the fiber $\pi^{-1}(x)$ is an elliptic curve or genus one algebraic curve. It is known that the number of topological types of elliptic Calabi–Yau threefolds is finite [13, 14]. Furthermore, it has been found that almost all known Calabi–Yau threefold constructions admit an elliptic fibration in some flop phase. For example, 99.3% (7837/7890) of the CICY threefolds have an “obvious” elliptic or genus one fibration [15], and all have at least one such fibration when $h^{1,1} \geq 4$. For the larger class of toric hypersurface Calabi–Yau threefolds, a systematic analysis was carried out in [16], finding that 99.994% of the threefolds in the Kreuzer–Skarke database have a manifest elliptic fiber in some phase (for each polytope there are many toric varieties, and different triangulations give different phases related under flops). A graph of the Hodge numbers of the Calabi–Yau threefolds in the Kreuzer–Skarke database is shown in Figure 1; the Hodge numbers of the 29,223 reflexive polytopes that give Calabi–Yau threefolds with no obvious elliptic fiber are shown in red.

2 Triple intersection numbers and topological equivalence

As described above, an important open question is to identify a systematic algorithm for determining when two sets of discrete data that characterize a topological Calabi–Yau three-
fold are equivalent under a basis change. In particular, we focus on the specific part of that problem associated with the triple intersection numbers

\[ C_{ijk} \in \mathbb{Z}, \quad i, j, k \leq N. \]  

These intersection numbers form a 3-index symmetric tensor so that \( C_{ijk} = C_{jik} = C_{ikj} \).

We wish to address the following mathematical question:

**Problem:** Given two 3-index symmetric tensors \( C, C' \) of degree \( N \) associated with Calabi–Yau threefold geometries, does there exist an integer linear change of basis \( \Lambda_{ij} \in \text{SL}(N, \mathbb{Z}) \) such that

\[ C'_{ijk} = \Lambda_{il} \Lambda_{jm} \Lambda_{kn} C_{lmn} \]  

Because the set of possible linear transformations \( \Lambda \) is infinite, there is no simple way to check the equivalence in finite time. In fact, there is no known algorithm to check this equivalence in finite time, whether or not it is known that the triple intersection numbers correspond to allowed Calabi–Yau threefold geometries.

Some partial results in this direction have been found. In particular, there are certain functions of the triple intersection numbers that are invariant under a basis change; thus, if the invariants differ between the intersection numbers \( C \) and \( C' \), there clearly cannot be an equivalence. Some of these invariants are summarized in [2]; for example, \( \gcd(\{C_{ijk}\}_{1 \leq i, j, k \leq N}) \) is clearly invariant under any transformation \( \Lambda_{ij} \in \text{SL}(N, \mathbb{Z}) \). Recently, some new invariants have been identified using the methodology of limiting mixed Hodge structures [17].

Because there is no known algorithm for checking this equivalence, this is a natural candidate for the application of machine learning (ML) methodology. In recent years, as elaborated in some detail in the recent text [3], there has been a great deal of work on using ML methods to solve discrete problems related to Calabi–Yau geometries, with varying degrees of success [18]–[33]. On one hand, one might hope that by presenting an ML system with a variety of data of triple intersection numbers with families of equivalence known from construction it may be possible to find some approximate method for checking equivalence. More ambitiously, one might hope to identify through this approach some systematic structure in the triple intersection numbers such as new invariants that may lead to an analytic solution of the problem.

Direct application of standard ML techniques to this problem does not immediately give insight, so we turn to a related problem to understand how machine learning can address discrete transformation problems of this type.
3 Machine learning and a simpler matrix problem

As an analogue of the problem of identifying when two 3-index tensors are equivalent under a basis change, we can consider the simpler problem of using machine learning to figure out when two matrices (i.e., 2-index tensors) are similarly equivalent. To relate this to some known results, we also drop for now the condition that the entries are integer. We can then consider the following simpler problem:

**Problem:** Given complex matrices $A_{ij}, B_{ij}$, does there exist a unitary transformation $\Lambda$ so that

$$B = \Lambda A \Lambda^\dagger? \tag{3}$$

There is a known set of criteria that determines when two such matrices are equivalent in this way. Specht’s theorem [34] states that $A \sim B$ iff

$$\begin{align*}
\text{Tr } A &= \text{Tr } B, \\
\text{Tr } A^2 &= \text{Tr } B^2, \\
\text{Tr } AA^\dagger &= \text{Tr } BB^\dagger, \\
\vdots
\end{align*} \tag{4}$$

i.e., $\text{Tr} W(A, A^\dagger) = \text{Tr} W(B, B^\dagger)$ for all words $W$ that are functions of the matrix and its adjoint.

In fact, for matrices of a given size $N$, it is sufficient to check the agreement between these invariants only up to a certain length. For $N = 2$, it is sufficient to check only up to words of degree 2 (i.e., 3 independent conditions). For $N = 3$, words up to degree 6 suffice (7 conditions). For general $N$ it has been proven that it is sufficient to check up to degree $N \sqrt{2N^2/(N-1) + 1/4} + N/2 - 2$, and it has been conjectured that the degree of sufficiency actually scales linearly in $N$ [35, 36].

Thus, we know that for matrices (2-index tensors), this is a solvable problem even when the entries are non-integer. Thus, it is perhaps illuminating to see how machine learning can do at learning to check such equivalences, and we can see if a simple machine learning system can be designed that can essentially “discover” Specht’s theorem.

Motivated by these considerations, we have implemented some simple ML algorithms to attempt to solve the matrix equivalence problem using the neural network functionality built in to Mathematica.

The simplest experiment is to try random real matrices $A, B$ with a simple multi-layer ReLU network, providing data in which half of the cases given are equivalent matrices and the other half are inequivalent, with no further constraints. This kind of network quickly
Figure 2: Network architecture for learning orthogonal equivalence of $3 \times 3$ real matrices with fixed first-order invariant. The input matrices $A$ and $B$ are fed into dot layers that compute the matrix squares, and then $A, A^2, B,$ and $B^2$ are fed into restOfNet, which is a multi-layer ReLU network.

Figure 3: Error rates during training for two different network architectures learning to identify orthogonal equivalence of $3 \times 3$ real matrices with fixed first-order invariant. On the left, a multi-layer ReLU network; on the right, the architecture shown in Figure 2 converges to near-100% accuracy. Some investigation, however, shows that this network is essentially just checking the first (linear) condition in (4), i.e., comparing the traces of the input matrices. It is not surprising that a simple ML system can rapidly detect when two matrices satisfy this simple linear condition.

The next challenge is to provide a data set in which we fix the condition $\text{Tr} A = \text{Tr} B$ by hand for all pairs. Using such a dataset of $3 \times 3$ matrices, and a basic multi-layer ReLU network, with some training the network accurately recognizes pairs that are equivalent about 75% of the time and pairs that are inequivalent about 60% of the time. This is not a very good success rate. By changing the network architecture to explicitly compute
all products of terms in the first layer by hand (see Figure 2), however, the success rates immediately increase to 99% for equivalent pairs and 90% for inequivalent pairs. The error rates during training for both network architectures are shown in Figure 3. Thus, with this change of architecture the neural network “discovers” the quadratic term in Specht’s theorem. It seems, however, that higher products must similarly be included by hand to efficiently converge on a system that knows about the additional constraints in (4). Similar results hold for complex matrices, where we must use a slightly more complicated version of the architecture shown in Figure 2 to include Tr AA† as well as Tr A².

4 Conclusions

In this brief extended abstract we have described a basic problem in identifying equivalence of symmetric 3-index tensors that is central to classifying Calabi–Yau threefolds by discrete information characterizing their topology. We have outlined the first steps of an effort to use machine learning to solve a simpler version of the problem with 2-index tensors where the analytic solution is known. Several lessons that can be taken from this are the following:

- Out of the box ML algorithms do not immediately yield insight in discrete problems of this kind.

- The fact that including product terms by hand in the ML architecture facilitates “discovery” by the neural network of nonlinear constraints suggests that networks with different basic functional units may perform dramatically differently on certain classes of problems; here we have an explicit understanding through Specht’s theorem of this structure.

- It has been pointed out by, e.g., [37] that in principle a general nonlinear layer type in neural networks can learn the multiplication function, so in principle the distinction between these different functional units may be essentially some factor of overhead. It seems likely, however, that the time to learn such things may be exponentially large so that in practice network architecture and the form of nonlinear function used may be crucial in getting networks to learn certain structures.

The work described here is really only a first step at using ML methods for analyzing this problem. It would be interesting to extend the analysis further to reproduce higher order terms in the matrix case and to investigate whether sufficiently sophisticated network architectures could learn to recognize equivalent 3-index integer tensors for the Calabi–Yau equivalence problem, and if so whether it would be possible to extract new analytic information such as new Calabi–Yau threefold invariants from this analysis.
Acknowledgments

We would like to thank Lara Anderson, James Gray, Harold Erbin, Tristan Hübsch, Max Tegmark, and Yinan Wang for helpful discussions. This work arose in the context of collaborative visits supported by a grant from the MISTI MIT-Africa-Imperial College seed fund initiative, and was supported in part by this grant. VJ is supported in part by the South African Research Chairs Initiative of the National Research Foundation, grant number 78554 and the Simons Foundation Mathematical and Physical Sciences Targeted Grants to Institutes, Award ID:509116. WT is also supported by DOE grant DE-SC00012567. APT is supported in part by the DOE (HEP) Award DE-SC0013528. This work was also supported in part by the National Science Foundation under Cooperative Agreement PHY-2019786.

References

[1] P. Candelas, G. T. Horowitz, A. Strominger and E. Witten, “Vacuum configurations for superstrings,” Nucl. Phys. B 258, 46 (1985).

[2] T. Hübsch, “Calabi–Yau Manifolds: A Bestiary for Physicists,” World Scientific, 1992.

[3] Y.-H. He, “The Calabi–Yau Landscape: From Geometry, to Physics, to Machine Learning,” Springer, Lecture Notes in Mathematics 2293, 2021.

[4] S.-T. Yau, “Open problems in geometry,” Proc. Symp. Pure Math 54, 1, 1-28 (1993).

[5] C. T. C. Wall, “Classification problems in differential topology. V,” Inventiones mathematicae 1, 355 (1966).

[6] V. Batyrev, “Variations of the mixed Hodge structure of affine hypersurfaces in algebraic tori,” Duke Math. Journ. 69, 349 (1993).

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

[8] M. Kreuzer and H. Skarke, http://hep.itp.tuwien.ac.at/~kreuzer/CY.html.

[9] P. Candelas, A. M. Dale, C. A. Lutken and R. Schimmrigk, “Complete intersection Calabi–Yau manifolds,” Nucl. Phys. B 298, 493 (1988).

[10] L. B. Anderson, F. Apruzzi, X. Gao, J. Gray and S. J. Lee, “A new construction of Calabi–Yau manifolds: Generalized CICYs,” Nucl. Phys. B 906, 441 (2016) arXiv:1507.03235 [hep-th].

[11] C. Vafa, “Evidence for F-Theory,” Nucl. Phys. B 469, 403 (1996) arXiv:hep-th/9602022.
[12] D. R. Morrison and C. Vafa, “Compactifications of F-Theory on Calabi–Yau threefolds – I,” Nucl. Phys. B 473, 74 (1996) arXiv:hep-th/9602114; “Compactifications of F-Theory on Calabi–Yau threefolds – II,” Nucl. Phys. B 476, 437 (1996) arXiv:hep-th/9603161.

[13] A. Grassi, “On minimal models of elliptic threefolds,” Math. Ann. 290 (1991) 287–301.

[14] M. Gross, “A finiteness theorem for elliptic Calabi–Yau threefolds,” Duke Math. Jour. 74, 271 (1994).

[15] L. B. Anderson, X. Gao, J. Gray and S. J. Lee, “Multiple fibrations in Calabi–Yau geometry and string dualities,” JHEP 1610, 105 (2016) arXiv:1608.07555 [hep-th]; “Fibrations in CICY threefolds,” JHEP 1710, 077 (2017) arXiv:1708.07907 [hep-th].

[16] Y. C. Huang and W. Taylor, “Fibration structure in toric hypersurface Calabi–Yau threefolds,” JHEP 03 (2020), 172 arXiv:1907.09482 [hep-th].

[17] T. W. Grimm, F. Ruehle and D. van de Heisteeg, “Classifying Calabi–Yau threefolds using infinite distance limits,” Commun. Math. Phys. 382 (2021) no.1, 239-275 arXiv:1910.02963 [hep-th].

[18] Y. H. He, “Deep-Learning the landscape,” arXiv:1706.02714 [hep-th].

[19] F. Ruehle, “Evolving neural networks with genetic algorithms to study the string landscape,” JHEP 08 (2017), 038 arXiv:1706.07024 [hep-th].

[20] J. Carifio, J. Halverson, D. Krioukov and B. D. Nelson, “Machine learning in the string landscape,” JHEP 09 (2017), 157 arXiv:1707.00655 [hep-th].

[21] Y. N. Wang and Z. Zhang, “Learning non-Higgsable gauge groups in 4D F-theory,” JHEP 08 (2018), 009 arXiv:1804.07296 [hep-th].

[22] K. Bull, Y. H. He, V. Jejjala and C. Mishra, “Machine learning CICY threefolds,” Phys. Lett. B 785 (2018), 65-72 arXiv:1806.03121 [hep-th].

[23] D. Klaewer and L. Schlechter, “Machine learning line Bundle cohomologies of hypersurfaces in toric varieties,” Phys. Lett. B 789 (2019), 438-443 arXiv:1809.02547 [hep-th].

[24] K. Bull, Y. H. He, V. Jejjala and C. Mishra, “Getting CICY high,” Phys. Lett. B 795, 700-706 (2019) arXiv:1903.03113 [hep-th].

[25] C. R. Brodie, A. Constantin, R. Deen and A. Lukas, “Machine learning line bundle cohomology,” Fortsch. Phys. 68 (2020) no.1, 1900087 arXiv:1906.08730 [hep-th].
[26] M. Bies, M. Cvetič, R. Donagi, L. Lin, M. Liu and F. Ruehle, “Machine learning and algebraic approaches towards complete matter spectra in 4d F-theory,” JHEP 01 (2021), 196 arXiv:2007.00009 [hep-th].

[27] H. Erbin and R. Finotello, “Inception neural network for complete intersection Calabi–Yau 3-folds,” Mach. Learn. Sci. Tech. 2 (2021) no.2, 02LT03 arXiv:2007.13379 [hep-th].

[28] H. Erbin and R. Finotello, “Machine learning for complete intersection Calabi–Yau manifolds: A methodological study,” Phys. Rev. D 103, no.12, 126014 (2021) arXiv:2007.15706 [hep-th].

[29] Y. H. He and A. Lukas, “Machine learning Calabi–Yau four-folds,” Phys. Lett. B 815 (2021), 136139 arXiv:2009.02544 [hep-th].

[30] H. Erbin, R. Finotello, R. Schneider and M. Tamaazousti, “Deep multi-task mining Calabi–Yau four-folds,” Mach. Learn. Sci. Tech. 3 (2022) no.1, 015006 arXiv:2108.02221 [hep-th].

[31] X. Gao and H. Zou, “Machine learning to the orientifold Calabi–Yau with string vacua,” arXiv:2112.04950 [hep-th].

[32] D. S. Berman, Y. H. He and E. Hirst, “Machine learning Calabi–Yau hypersurfaces,” arXiv:2112.06350 [hep-th].

[33] P. Berglund, B. Campbell and V. Jejjala, “Machine learning Kreuzer–Skarke Calabi–Yau threefolds,” arXiv:2112.09117 [hep-th].

[34] W. Specht, “Zur theorie der matrizen. II.,” Jahresbericht der Deutschen Mathematiker-Vereinigung 50, 19 (1940).

[35] A. Paz, “An application of the Cayley–Hamilton theorem to matrix polynomials in several variables,” Linear and Multilinear Algebra 15, 161 (1984).

[36] A. Freedman, R. Gupta and R. Guralnick, “Shirshov’s theorem and representations of semigroups,” Pacific J. Math. 181, 159, (1997).

[37] H. W. Lin, M. Tegmark and D. Rolnick, “Why does deep and cheap learning work so well?,” J. Stat. Phys. 168, 1223 (2017) arXiv:1608.08225 [cond-mat].