Generating Calabi-Yau manifolds with transformers

Charles Arnal[*]
Meta AI

Jacky H. T. Yip[*]
Department of Physics
University of Wisconsin-Madison

François Charton
Axiom
Ecole nationale des Ponts et Chaussées

Gary Shiu
Department of Physics
University of Wisconsin-Madison

Abstract

Triangulations, i.e., well-structured decompositions of geometric objects into triangle-like pieces, are central objects in many domains of mathematics and physics. In particular, fine, regular, and star triangulations (FRSTs) of 4D reflexive polytopes give rise to smooth Calabi-Yau threefolds, which are of significant interest in string theory. However, the high dimensionality and combinatorial complexity of triangulations make them particularly challenging to model with classical numerical methods or machine learning. In this work, we show that transformers, equipped with an appropriate encoding scheme, can be effectively trained to representatively generate new FRSTs across a range of polytope sizes. This opens the door to both concrete applications to the classification of Calabi-Yau manifolds and further research in physics, combinatorics and algebraic geometry.

1 Introduction

While machine learning has been applied to great success to the study of graphs (1; 2; 3; 4; 5; 6; 7; 8), higher dimensional geometric structures remain underexplored. In particular, *triangulations*–well-structured decomposition of geometric objects into triangle-like pieces called *simplices*–have scarcely been studied (9; 10), despite their key role in several domains of mathematics and physics, such as combinatorics (11; 12; 13; 14), algebraic geometry (15; 16; 17), topological data analysis (18; 19) and string theory (20; 21; 22). Human researchers often struggle to visualize and manipulate higher dimensional triangulations due to the limitations of human visual intuition; consequently, many problems which require defining special triangulations with desirable properties remain open. Machine learning methods do not face the same limitations and might prove successful at modelling interesting distributions of triangulations, as has been the case for other classes of combinatorial objects (23; 24).

String theory is a particularly promising area of application. Understanding the string landscape–determining string theories that are compatible with realistic particle physics and gravity–is a challenging task because of the number of possible choices for compactification: packing the many dimensions of string theory into the four dimensions of physical space. Fine, regular, star triangulations (FRSTs) of 4D reflexive polytopes give rise to toric Calabi-Yau threefolds, which in turn provide 4D vacuum configurations of superstring theory that can accommodate realistic physics (25; 26). As the number of 4D reflexive polytopes is large but finite–473,800,776 in total (27)–all possible configurations could theoretically be enumerated by considering all the possible triangulations of each of these polytopes. The number of FRSTs grows exponentially with the number of vertices in the polytope, which makes this strategy computationally prohibitive. A more reasonable strategy is to

[*]These authors contributed equally.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

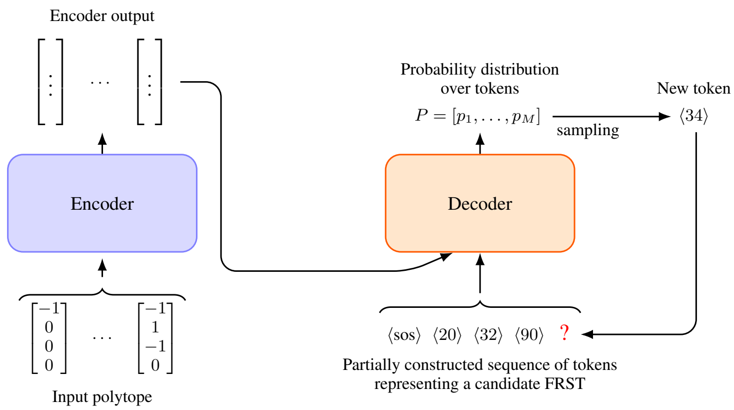

Figure 1: The high level pipeline for our CYTransformer model, represented here in inference mode.

develop efficient techniques for generating diverse FRSTs for large polytopes. Current approaches involving non-learning algorithms (28) do not scale with the number of vertices. Genetic (29) and reinforcement learning (30) techniques do not easily transfer from one polytope to another. These limitations can be attributed to the combinatorially explosive nature of the space of triangulations, as well as to the difficulty in finding an amenable encoding of triangulations.

In this paper, we propose a generative technique for FRSTs based on encoder-decoder transformers (31), which we call *CYTransformer*. The model is trained on known FRSTs, encoded as sequences of tokens representing simplices, to predict triangulations of input polytopes encoded as sequences of coordinates describing their vertices. Whereas model outputs are not guaranteed to be FRSTs (or even proper triangulations), they can be verified at low cost using problem-specific tools, such as `CYTools` (32). We show that once trained on a particular set of polytopes and their FRSTs, CYTransformers can generate new FRSTs of yet unseen polytopes that are representative of the full ensemble of FRSTs.

2 Toric Calabi-Yau manifolds and Fine Regular Star Triangulations (FRSTs)

2.1 Triangulations

A d-dimensional *simplex* in $\mathbb{R}^n$ is the convex hull of $d+1$ affinely independent points in $\mathbb{R}^n$–segments and triangles are examples of 1D and 2D simplices respectively. Given an n-dimensional polytope $\Delta \subset \mathbb{R}^n$ (i.e., the convex hull of finitely many points that affinely span $\mathbb{R}^n$), a *triangulation* of Δ is a set of n-dimensional simplices $S_1, \ldots, S_k \subset \Delta$ such that their union is Δ, and such that any two distinct simplices S_i, S_j either do not intersect or intersect along a common face. One case of particular interest is when the triangulation has to be supported on the vertices of the polytope, i.e., when the vertices of any of its simplices have to be vertices of Δ (as opposed to interior points). An illustration is given in Figure 2.

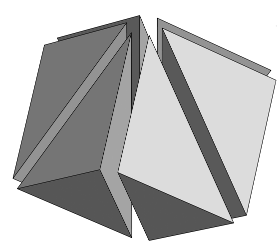

Figure 2: A triangulation of a 3D cube (from (33)).

Triangulations are among the most frequent higher-dimensional geometric and combinatorial objects in mathematics, be it in topology, geometry, algebraic geometry or combinatorics. In some regards, they can be seen as higher-dimensional analogues of graphs. Despite their ubiquitousness, they are difficult to handle by either human intuition or automated methods. This is in part due to the number of possible triangulations of a polytope typically growing exponentially fast in the number of its vertices. Furthermore, while triangulations are very numerous, they are also very rare among all the possible collections of sub-simplices of a given polytope, due to the rigid union and intersection conditions given above, which often makes them impossible to enumerate using brute force methods.

2.2 Calabi-Yau manifolds, reflexive polytopes, and FRSTs

As mentioned in the introduction, Calabi-Yau manifolds play a crucial role in string theory. They are smooth and compact spaces equipped with a Ricci-flat Kähler metric, and are prime candidates for string compactification: the Ricci-flatness preserves supersymmetry, while the toric structure endows these spaces with torus symmetries needed in particle physics model building (25; 34).

One of the ways to define toric Calabi-Yau manifolds is with *reflexive polytopes* and their *fine, regular, and star triangulations (FRSTs)*. A reflexive polytope Δ is the convex hull of a finite set of points of $\mathbb{Z}^n$ such that its dual polytope $\Delta^\circ := \mathrm{Conv}(\{y \in \mathbb{Z}^n \mid \langle m, y\rangle \geq -1\ \forall m \in \Delta\})$ is also the convex hull of a finite set of points of $\mathbb{Z}^n$. It can be shown that the interior of a reflexive polytope contains exactly one point of $\mathbb{Z}^n$, which is taken to be the origin. The set of points consisting of all points in $\Delta \cap \mathbb{Z}^n$, excluding those strictly interior to codimension-1 faces, together with the origin, is referred to as the set of resolved vertices of Δ. Up to lattice automorphisms in $GL(n, \mathbb{Z})$, the number of reflexive polytopes in any given dimension n is finite. In the remainder of this paper, we split and organize reflexive polytopes according to the number N_{vert} of their vertices.[1]

Given a reflexive polytope, we are interested in special triangulations of its resolved vertices (i.e., triangulations such that the vertices of their simplices are resolved vertices). Those triangulations $\mathcal{T}$ must satisfy: every resolved vertex appears as a vertex of some simplex in $\mathcal{T}$ (**Fine (F)**), $\mathcal{T}$ arises as the projection of the lower faces of a convex polytope constructed from lifting each resolved vertex p_i to $(p_i, h_i) \in \mathbb{Z}^n \times \mathbb{R}$ with some height $h_i \in \mathbb{R}$ (**Regular (R)**), and every simplex in $\mathcal{T}$ contains the origin as a vertex (**Star (S)**). Hence we call these FRSTs. Given a reflexive polytope Δ and an FRST $\mathcal{T}$ of Δ, a classical algebraic construction yields a smooth, projective, and compact Calabi-Yau threefold, which is of special interest in string theory (20; 35).

3 Generating Calabi-Yau manifolds with transformers

3.1 Encoding polytopes and triangulations

We encode polytopes as sequences of 4D vectors representing their vertices (with the origin omitted). For a fixed N_{vert}, we encode collections of simplices of polytopes (and in particular FRSTs) with N_{vert} vertices as tokens. To do so, we fix a mapping F between the subsets of cardinality 4 of $\{1, \ldots, N_{\text{vert}} - 1\}$ and the integers $\{1, \ldots, \binom{N_{\text{vert}}-1}{4}\}$. For a given input polytope Δ with vertices $(V_1, \ldots, V_{N_{\text{vert}}-1})$, we then represent a star simplex $\{\text{origin}, V_{i_1}, V_{i_2}, V_{i_3}, V_{i_4}\}$ (where $1 \leq i_1 < i_2 < i_3 < i_4 \leq N_{\text{vert}} - 1$) by the image by F of the set (i_1, i_2, i_3, i_4). As a consequence, the meaning of each of these tokens is always conditioned not only on the input polytope, but also on the order in which the vertices of the input polytope appear in its encoding. We also use special begin-of-sequence, end-of-sequence, and padding tokens.

3.2 Model architecture

CYTransformer adopts the encoder-decoder transformer architecture introduced in the landmark paper (31). The high-level architecture, which we also illustrate in Figure 1, is as follows: at any given step, the model takes as input the entire input polytope and a partially constructed sequence of tokens. Each token represents a simplex. The input polytope is processed by the encoder to produce a sequence of high-dimensional vectors: the encoder output. The encoder output, together with the sequence of tokens generated so far, are used as inputs by the decoder. The decoder then outputs a vector $\boldsymbol{P}$ representing a probability distribution over the set of all tokens considered. At inference time, a new token is sampled from this probability distribution, and added to the sequence being constructed. This process is repeated until the special end of sequence token is produced, or until the sequence reaches a predefined length. At training time, the parameters of the encoder and the decoder are trained to minimize the cross-entropy loss between the vector $\boldsymbol{P}$ and the ground truth from the training sequence. Both during training and at inference time, we apply a form of data augmentation by randomly permuting the vertices of the input polytopes.

3.3 Training data and validation

`CYTools` (32) is a recently developed software package, with polytope triangulation as one of its primary design focuses. It can fetch reflexive polytopes from the complete list of the Kreuzer-Skarke

[1] One could also consider the $(1, 1)$-Hodge number $h^{1,1}$ of the corresponding Calabi-Yau manifolds, which is of greater interest to physicists. To simplify the presentation of our work, we restrict ourselves to polytopes such that $N_{\text{vert}} = h^{1,1} + 5$, which is the case of most polytopes with small number of vertices.

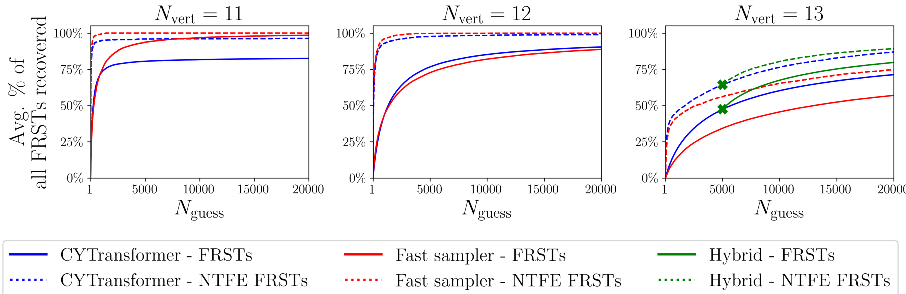

Figure 3: **Comparison of FRST recovery performance.** The average percentages of distinct FRSTs (solid) and distinct FRSTs that are not two-face equivalent (dashed) recovered by CYTransformer (blue), the fast sampler (red) and a hybrid method (green) as a function of the number of samples N_{guess} are shown across 200 test polytopes, for various N_{vert}.

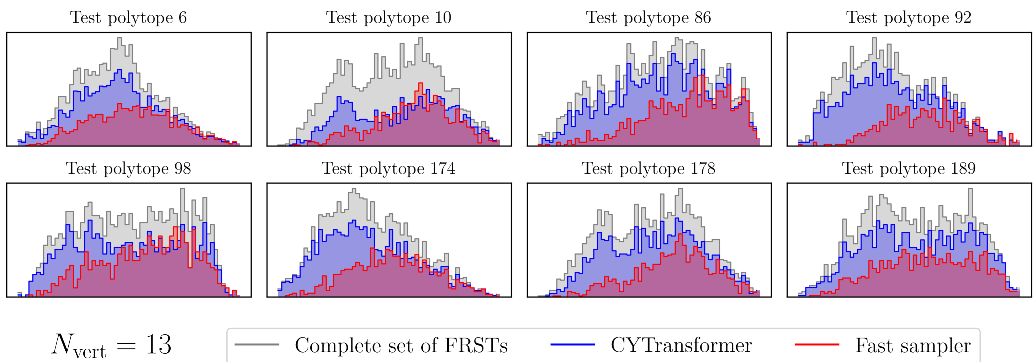

Figure 4: **Sampling distributions.** Height vector similarity score histograms are shown for 8 test polytopes with $N_{\text{vert}} = 13$. For each polytope, we compare CYTransformer (blue) and the fast sampler (red) along wih the full population distribution (grey).

database (27), return FRSTs for a given polytope following various distributions, and check whether a collection of simplices of a given polytope is an FRST. We use `CYTools` to generate our initial training data and to check whether the candidate FRSTs generated by our trained models are valid. We also use its FRSTs generating algorithms as a baseline to which we compare our trained models.

4 Experiments and results

We train CYTransformers for $N_{\text{vert}} \in \{10, 11, 12, 13\}$, and find that the trained models are capable of generating FRSTs that were not part of their training set, including for input polytopes on which they were never trained. This is remarkable, considering the very strict set of conditions that defines FRSTs and their scarcity among sequences of simplices; note that it would typically take hours for a human expert to find an FRST of a small reflexive polytope without using specialized software.

Furthermore, we observe that CYTransformers are competitive with pre-existing methods. We sample candidate FRSTs from a trained CYTransformer as well as using `CYTools`'s fast sampler algorithm, and compare in Figure 3 the number of unique FRSTs obtained with either method as a function of the number of guesses. For small configurations, the FRST space is small, allowing the fast sampler to efficiently scan the space via random perturbations. By contrast, CYTransformer outperforms the fast sampler for larger polytopes by capturing the global distribution of FRSTs more effectively, while the fast sampler remains a local method, confined to a comparatively narrow subset of triangulations. We also test for $N_{\text{vert}} = 13$ a hybrid method, where outputs from CYTransformer are used as seeds for the fast sampler algorithm, which is shown to perform even better.

We also compare CYTransformer and the fast sampler using height vector-based similarity score histograms for test polytopes with $N_{\text{vert}} = 13$, and in Figure 4 we show 8 to highlight clear differences in sampling behavior. We find that the CYTransformer's histograms consistently (across the 200 test polytopes) match the shape of the full population distribution, while the fast sampler's histograms often exhibit skewed profiles. This indicates that although both methods can eventually cover much of the FRST space, they do so differently: the CYTransformer samples in proportion to the true density of FRSTs, while the fast sampler tends to concentrate on particular regions.

Acknowledgments and Disclosure of Funding

The authors thank Alessandro Mininno and Jakob Moritz for valuable discussions, and Moritz Münchmeyer for providing access to his research group's computing cluster. The work of G.S. and J.H.T.Y. is supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics under Award Numbers DE-SC-0023719 and DE-SC-0017647. F.C. was affiliated with Meta FAIR during the period when the main part of this research was conducted.

References

[1] T.N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, 2016.

[2] W.L. Hamilton, R. Ying and J. Leskovec, *Inductive representation learning on large graphs*, 2017.

[3] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, *Graph attention networks*, 2017.

[4] B. Perozzi, R. Al-Rfou and S. Skiena, *Deepwalk: online learning of social representations*, in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '14, p. 701–710, ACM, Aug., 2014, DOI.

[5] A. Grover and J. Leskovec, *node2vec: Scalable feature learning for networks*, 2016.

[6] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals and G.E. Dahl, *Neural message passing for quantum chemistry*, 2017.

[7] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu et al., *Graph neural networks: A review of methods and applications*, 2018.

[8] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and P.S. Yu, *A comprehensive survey on graph neural networks*, *IEEE Transactions on Neural Networks and Learning Systems* **32** (2021) 4–24.

[9] N. Sharp and M. Ovsjanikov, *Pointtrinet: Learned triangulation of 3d point sets*, 2020.

[10] H. Lei, R. Leng, L. Zheng and H. Li, *Circnet: Meshing 3d point clouds with circumcenter detection*, 2023.

[11] F. Luo and R. Stong, *Combinatorics of triangulations of 3-manifolds*, *Transactions of the American Mathematical Society* **337** (1993) 891.

[12] W.D. Neumann, *Combinatorics of triangulations and the chern-simons invariant for hyperbolic 3-manifolds*, in *Topology '90*, B. Apanasov, W.D. Neumann, A.W. Reid and L. Siebenmann, eds., (Berlin, Boston), pp. 243–272, De Gruyter (1992), DOI.

[13] M. Lackenby, *Taut ideal triangulations of 3-manifolds.*, *Geometry and Topology* **4** (2000) 369.

[14] K. Huszár, J. Spreer and U. Wagner, *On the treewidth of triangulated 3-manifolds*, *Journal of Computational Geometry* (2019) Vol. 10 No. 2 (2019): Special Issue of Selected Papers from SoCG 2018.

[15] I. Itenberg and O. Viro, *Patchworking algebraic curves disproves the ragsdale conjecture*, *The Mathematical Intelligencer* **18** (1996) 19.

[16] O. Viro, *Patchworking real algebraic varieties*, 2006.

[17] C. Arnal, *Patchworking real algebraic hypersurfaces with asymptotically large betti numbers*, *Journal of Topology* **15** (2022) 1154–1216.

[18] F. Chazal and B. Michel, *An introduction to topological data analysis: Fundamental and practical aspects for data scientists*, *Frontiers in Artificial Intelligence* **4** (2017) .

[19] G. Carlsson and M. Vejdemo-Johansson, *Topological Data Analysis with Applications*, Cambridge University Press (2021).

[20] V.V. Batyrev, *Dual polyhedra and mirror symmetry for Calabi-Yau hypersurfaces in toric varieties*, *J. Alg. Geom.* **3** (1994) 493 [alg-geom/9310003].

[21] M. Kreuzer, *Strings on calabi—yau spaces and toric geometry*, *Nuclear Physics B - Proceedings Supplements* **102–103** (2001) 87–93.

[22] R. Altman, J. Gray, Y.-H. He, V. Jejjala and B.D. Nelson, *A Calabi-Yau Database: Threefolds Constructed from the Kreuzer-Skarke List*, *JHEP* **02** (2015) 158 [1411.1418].

[23] A.Z. Wagner, *Constructions in combinatorics via neural networks*, 2021.

[24] A. Novikov, N. Vũ, M. Eisenberger, E. Dupont, P.-S. Huang, A.Z. Wagner et al., *Alphaevolve: A coding agent for scientific and algorithmic discovery*, 2025.

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

[26] F. Marchesano, G. Shiu and T. Weigand, *The Standard Model from String Theory: What Have We Learned?*, *Ann. Rev. Nucl. Part. Sci.* **74** (2024) 113 [2401.01939].

[27] M. Kreuzer and H. Skarke, *Complete classification of reflexive polyhedra in four-dimensions*, *Adv. Theor. Math. Phys.* **4** (2000) 1209 [hep-th/0002240].

[28] J.A. De Loera, J. Rambau and F. Santos, *Triangulations: Structures for Algorithms and Applications*, Springer Publishing Company, Incorporated, 1st ed. (2010).

[29] N. MacFadden, A. Schachner and E. Sheridan, *The DNA of Calabi-Yau Hypersurfaces*, 2405.08871.

[30] P. Berglund, G. Butbaia, Y.-H. He, E. Heyes, E. Hirst and V. Jejjala, *Generating triangulations and fibrations with reinforcement learning*, *Phys. Lett. B* **860** (2025) 139158 [2405.21017].

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez et al., *Attention Is All You Need*, in *31st International Conference on Neural Information Processing Systems*, 6, 2017 [1706.03762].

[32] M. Demirtas, A. Rios-Tascon and L. McAllister, *CYTools: A Software Package for Analyzing Calabi-Yau Manifolds*, 2211.03823.

[33] M. Kerber, R.F. Tichy and M. Weitzer, *Constrained triangulations, volumes of polytopes, and unit equations*, in *International Symposium on Computational Geometry*, 2016, https://api.semanticscholar.org/CorpusID:44417681.

[34] S. Katz, A. Klemm and C. Vafa, *Geometric engineering of quantum field theories*, *Nuclear Physics B* **497** (1997) 173–195.

[35] M. Demirtas, L. McAllister and A. Rios-Tascon, *Bounding the Kreuzer-Skarke Landscape*, *Fortsch. Phys.* **68** (2020) 2000086 [2008.01730].