

Transforming Calabi-Yau Constructions

Polytope Triangulation with Transformers

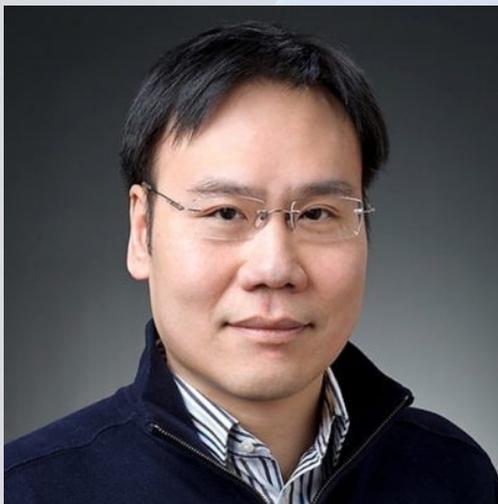
Jacky H. T. Yip

University of Wisconsin-Madison

hyip2@wisc.edu

Lepton Photon, Madison, August 2025

Based on 2507.03732, in collaboration with



Gary Shiu

University of Wisconsin-Madison



François Charton

FAIR, Meta
École des Ponts



Charles Arnal

FAIR, Meta

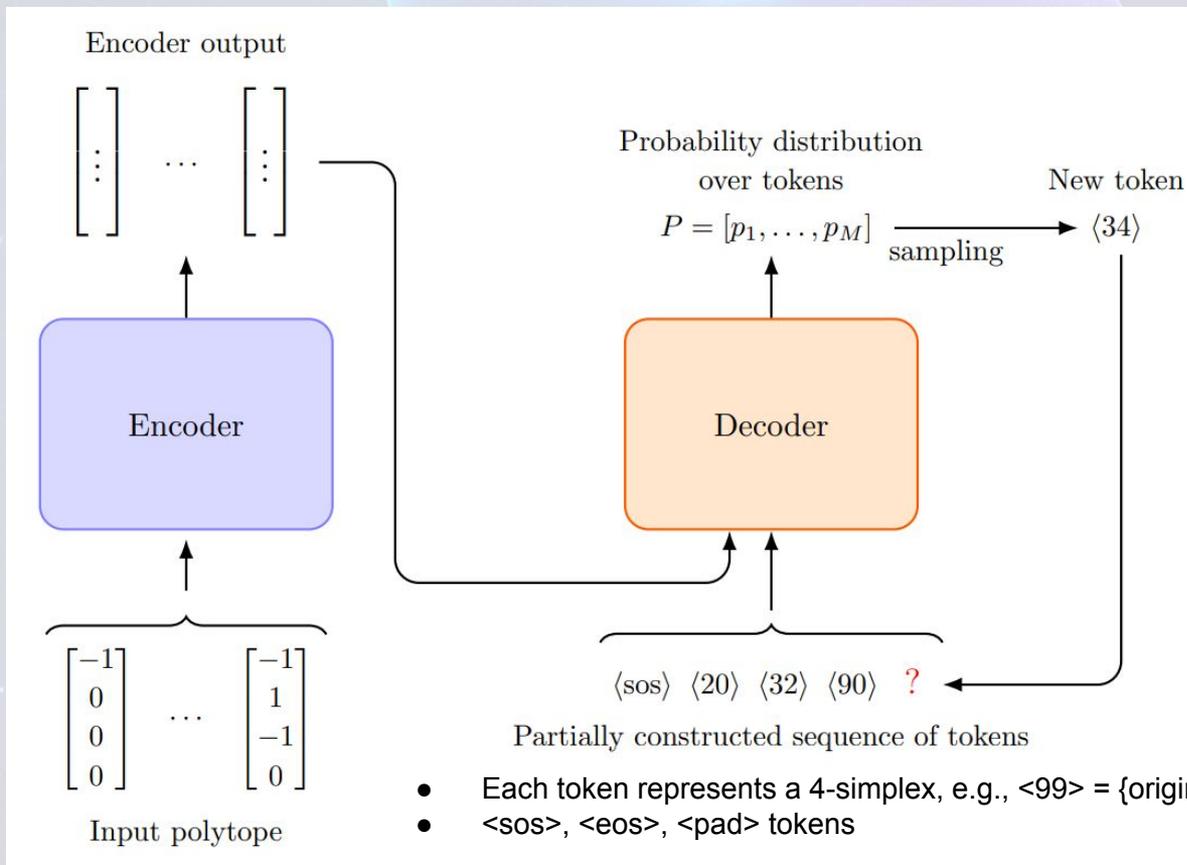
Toric Calabi-Yau manifolds from FRSTs of reflexive polytopes

- Compactify 10D (super)string theories on Calabi-Yau 3folds (CY3) to obtain 4D effective theories
- Batyrev's construction:
 - A triangulation of a 4D reflexive polytope defines a fan
 - The fan defines a toric variety
 - A hypersurface in this toric variety, defined using the dual polytope, gives a CY3
- That is, **constructing CY3 in this setting reduces to finding valid triangulations of polytopes**
- We require the triangulation to be **Fine, Regular, and Star (FRST)** so that the CY3 is smooth, Kähler, and compact:
 - **Fine:** Use lattice points too, except those strictly interior to codimension-1 faces
 - Resolved vertices: $N_{\text{vert}} = N + 1$; “polytope size”
 - **Regular:** Triangulation is the projection of the lower convex hull of lifted vertices
 - **Star:** All 4-simplices in the triangulation contain the origin as a vertex
- For convenience, we use only favorable polytopes such that $h^{1,1} = N_{\text{vert}} - 4 - 1$
 - No loss of generality, model cares only about the number of input vertices

Machine learning FRSTs and the transformer architecture

- Why machine learning?
 - Seeded non-learning random-walk samplers do not scale well with polytope size
 - A learning algorithm that **truly captures the construction rules** is expected to be **unbiased** and **generalizes** across different polytopes of different sizes
 - Opens door to **targeted search**
- Machine learning models
 - Hypergraph: Nodes as vertices and hyperedges as simplices
 - Unclear how to make it generative
 - **Transformer: Autoregressive token-by-token inference**
 - Each token represents a 4-simplex
 - Generative by design
 - Subsequent 4-simplices are conditioned on previous predicted ones

CYTransformer - The encoder-decoder architecture



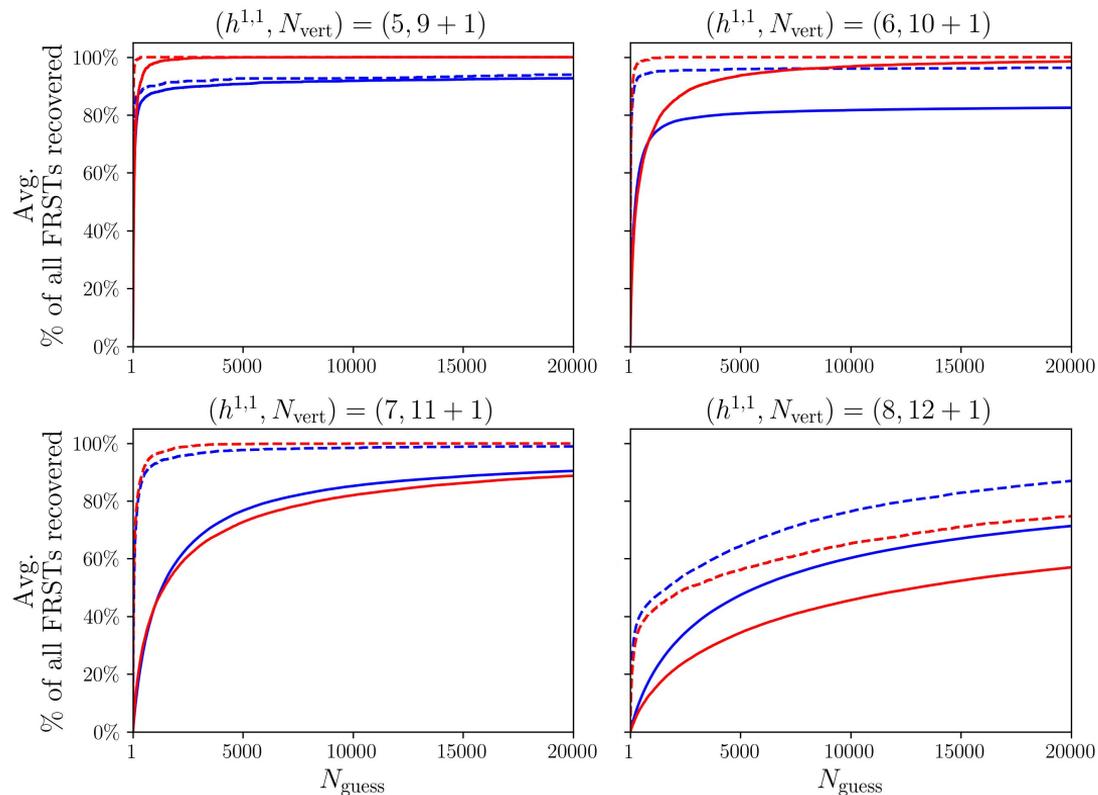
Hyperparameters:
16 attention layers
<120M learnable parameters

Training:
On <10,000 polytopes,
all FRSTs per polytope
2-3 days on 8 GPUs in parallel

Baseline for comparison - Fast sampler

- A simple non-learning method built into CYTools
- Generate triangulations by
 - Using the Delaunay height vector as seed
 - Perturb by adding noise sampled from a zero-mean Gaussian with a width
 - The width is fine-tuned for best performance
- Cheap to run

CYTransformer vs fast sampler - Recovery curves



- Averaged over 200 test polytopes
- For **small** polytopes, **fast sampler** performs well
 - Fast sampler scans small FRST space efficiently and thoroughly
- For **large** polytopes, **CYTransformer** clearly outperforms
 - CYTransformer explores the FRST space unbiasedly
- $(7, 11+1)$ is where machine learning beats brute-force randomness
- **Fast sampler is a local scanner; CYTransformer is a global explorer**

— CYTransformer - Counting distinct FRSTs

— Fast sampler - Counting distinct FRSTs

⋯ CYTransformer - Counting distinct NTFE FRSTs

⋯ Fast sampler - Counting distinct NTFE FRSTs

CYTransformer - Height vector similarity score

- Mapping from height vector to FRST is **many-to-one**
 - Adding affine functions of the coordinates to the height vector generates the same FRST
- **Remove affine ambiguity** by projecting a height vector onto the orthogonal complement of the subspace of affine functions

$$h_{\text{proj}} = h - h_{\text{aff}} = h - Pc$$

$$P = \begin{bmatrix} 1 & p_1^T \\ 1 & p_2^T \\ 1 & p_3^T \\ \vdots & \vdots \\ 1 & p_{N_{\text{vert}}}^T \end{bmatrix}$$

Finding coefficients in c such that

$$\arg \min_{c \in \mathbb{R}^5} |Pc - h|^2$$

The solution is

$$c = (P^T P)^{-1} P^T h$$

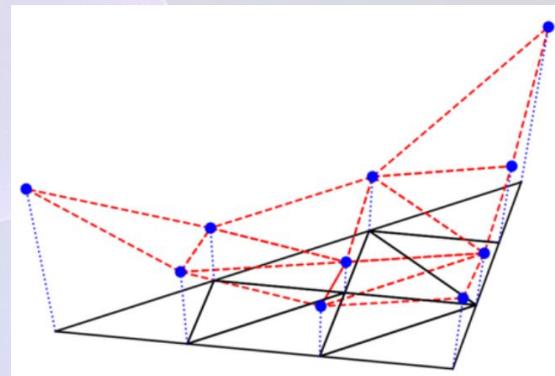


Figure from 2008.01730

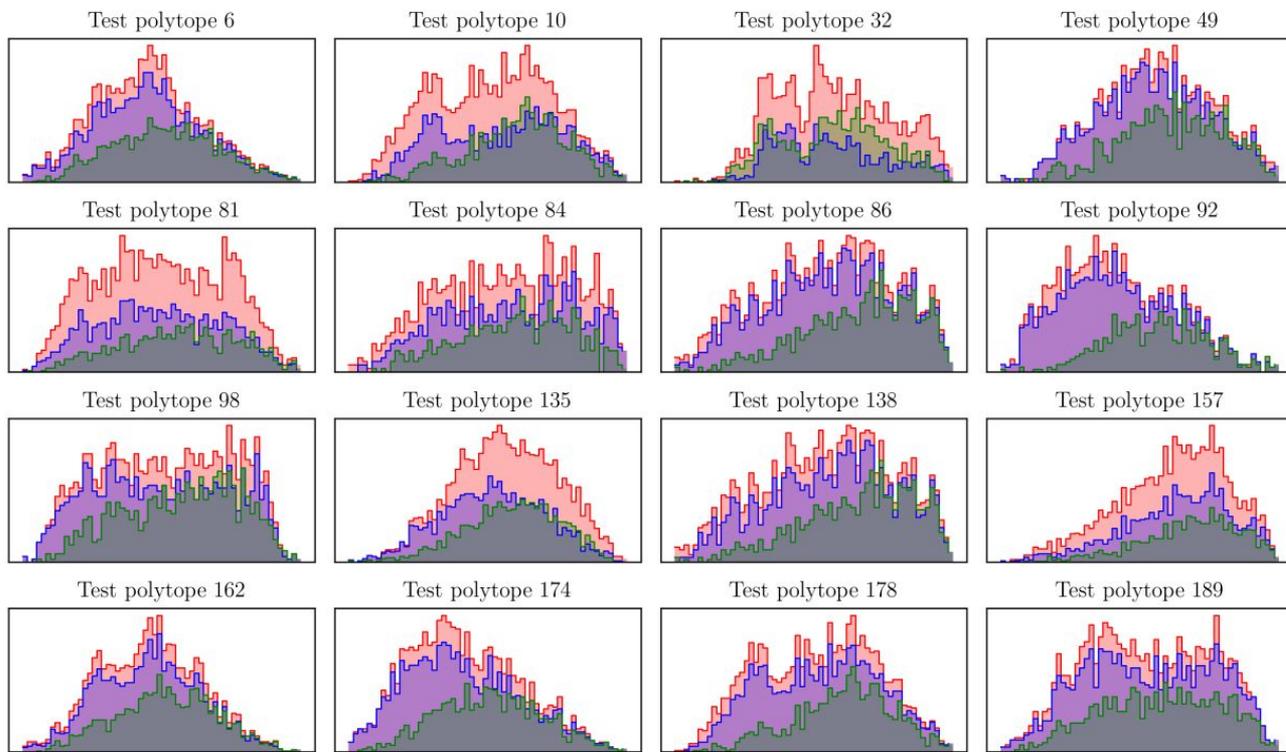
- Scaling projected height vector gives same FRST
 - Use the **scale-invariant** cosine similarity relative to the Delaunay height vector

$$\text{similarity score} = \frac{\text{cosine similarity}(h_{\text{proj}}, h_{\text{Delaunay}})}{h_{\text{Delaunay}}}$$

$$h_i = |p_i|^2$$

CYTransformer vs fast sampler - Height vector similarity score histogram

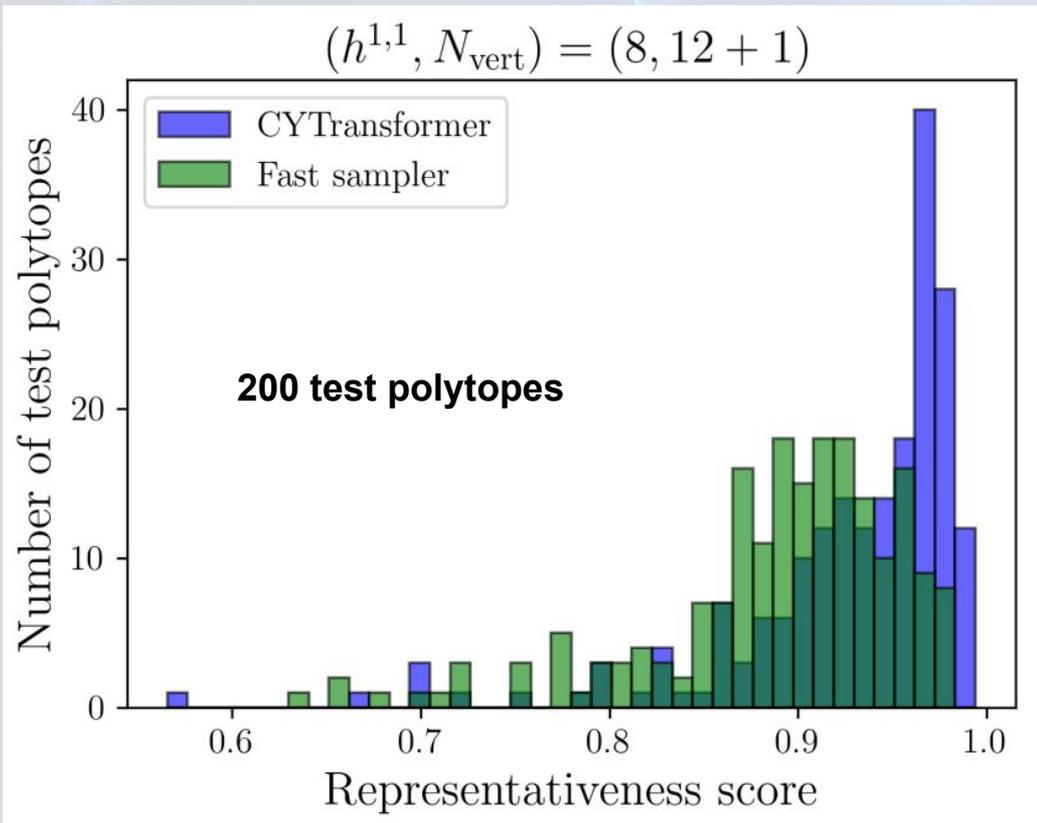
$$(h^{1,1}, N_{\text{vert}}) = (8, 12 + 1)$$



— CYTransformer
— Fast sampler
— Complete set

- Fast sampler does not match the population distribution - **biased sampling**
- We can **quantify the representativeness** by computing the cosine similarity between model histogram and population histogram

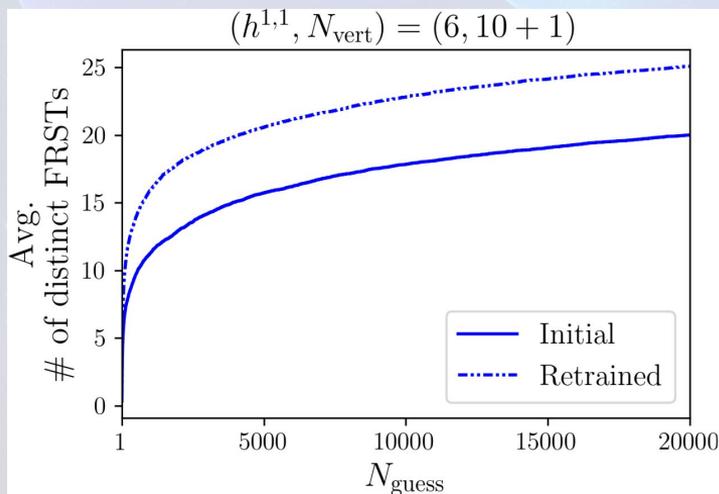
CYTransformer vs fast sampler - Representativeness



- CYTransformer
 - Peak near 1 with low variance
 - consistent representative sampling
- Fast sampler
 - Lower, more spread out scores
 - Inconsistent biased sampling

CYTransformer - Self-improvement capability

- A form of reinforcement learning given limited training data
 - a. Start with a small initial training set
 - b. Train CYTransformer on training set
 - c. Use the trained model to generate triangulations on the training polytopes
 - d. Add valid FRSTs to training set
 - e. Repeat b to d



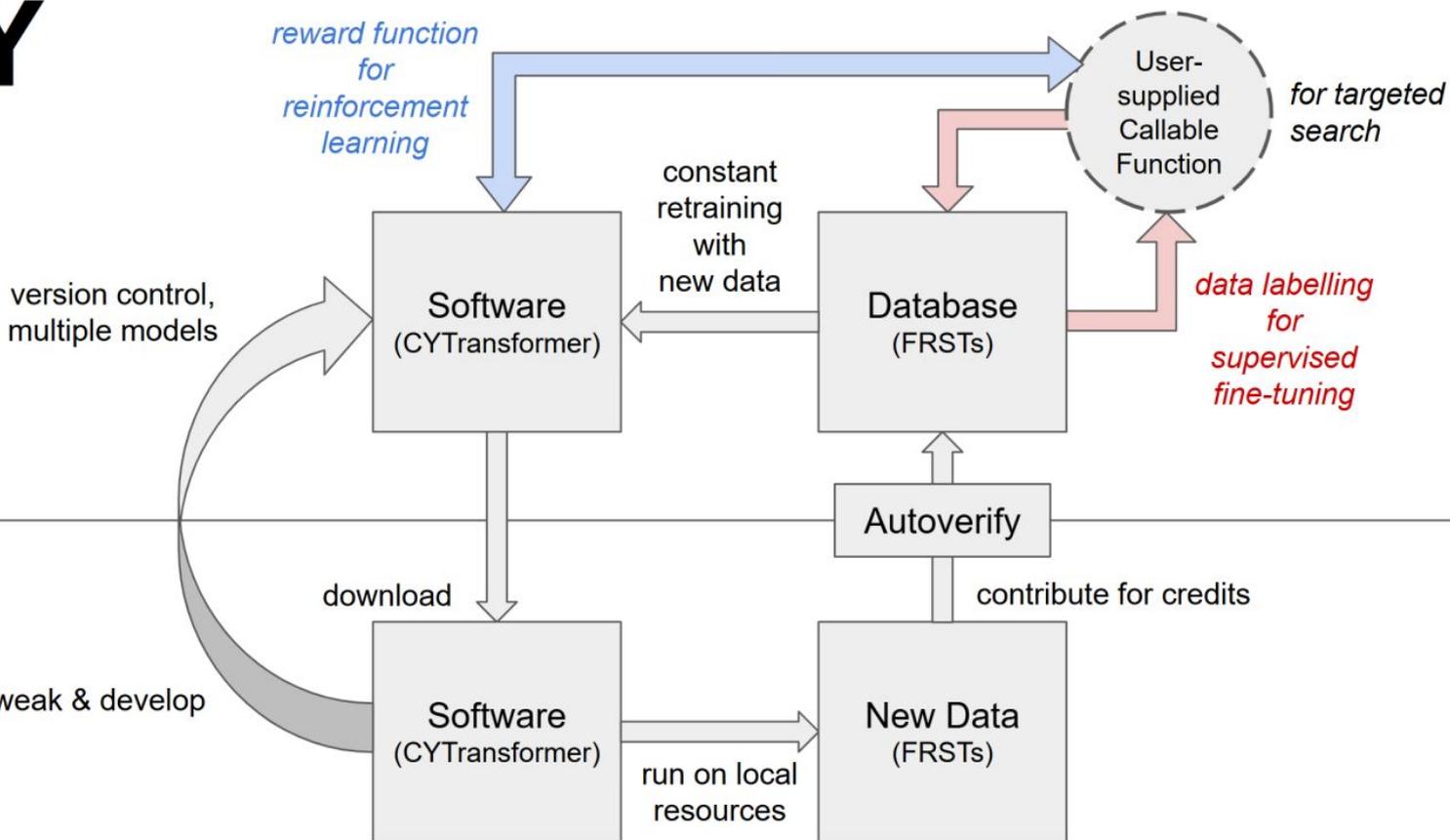
- Initial training set
 - 8000 polytopes
 - **5 FRSTs per polytope**
- Iteration 1
 - 8000 polytopes
 - **6.6 FRSTs per polytope**
- **25% increased performance** averaged over 200 test polytopes

Scaling it up - Communal involvement

- CYTransformer: Lessons learned and what to expect
 - ML models can generate **unbiased samples**, and **self-improve** on these samples
 - Techniques like **priming** will enable reliable generalization
 - **Targeted search** can be implemented **without changing model architecture**
- To realize these bootstrapping potentials, we need **constantly evolving models and datasets**
- This cannot be done without you, because
 - What if we have only one telescope for surveying galaxies?
 - Sure, I can keep it running for a long time to collect enough data to shrink error bars
 - But... more telescopes for simultaneously more data, more than one type of telescope...
- We need a **community-driven infrastructure** for such a **large-scale problem**
 - Publicly shared models and data grow together in a continuous and automated manner

AICY - AI-enabled living Calabi-Yau repositories

AICY



Server side

User side

AICY

AI-enabled living Calabi-Yau repositories

[Home](#)

[About](#)

[Sign Up](#)

[Software](#)

[Data](#)

[Collaborate](#)

[Contact](#)

[Cite](#)



Welcome to **AICY**.

Built for navigating and cataloging the Calabi-Yau landscape.
Powered by machine learning. Guided by fundamental physics.
Community-driven. Ever-evolving.

[Sign up for updates & beta access](#)

AICY

AI-enabled living Calabi-Yau repositories

[Home](#)

[About](#)

[Sign Up](#)

[Software](#)

[Data](#)

[Collaborate](#)

[Contact](#)

[Cite](#)



Sign up for updates & beta access

You'll receive updates and beta access information by email as we move toward the AICY launch.

Name

Affiliation

ORCID iD (optional)

Email

Submit

AICY

AI-enabled living Calabi-Yau repositories

[Home](#)

[About](#)

[Sign Up](#)

[Software](#)

[Data](#)

[Collaborate](#)

[Contact](#)

[Cite](#)



Propose a collaboration or project

The AICY team is always open to collaborations that expand, test, or apply the platform in new directions. We value your input and will respond at our earliest convenience.

Name

Affiliation

Email

Tell us more!

Briefly describe your idea, project goals, or how you'd like to collaborate.

Thank you very much!

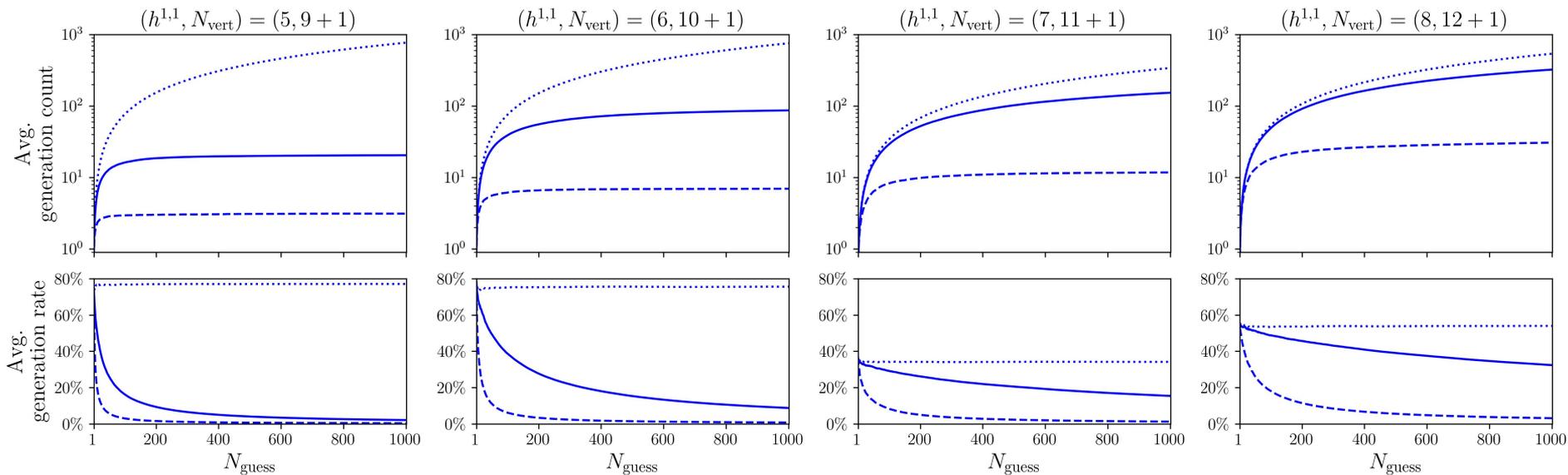
Paper:
2507.03732



AICY:



CYTransformer - Generation count and rate

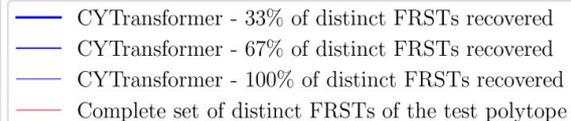
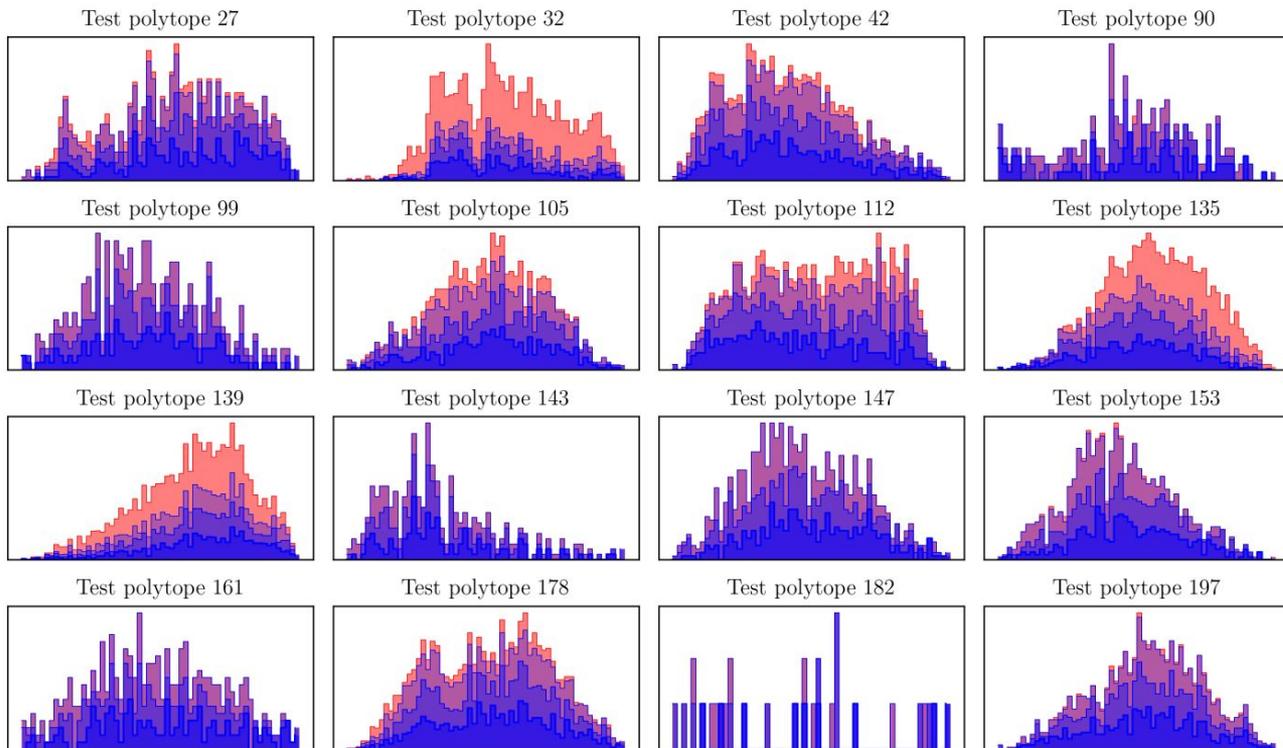


..... Counting all FRSTs — Counting distinct FRSTs - - - Counting distinct NTFE FRSTs

- Cumulative count per polytope averaged over 200 test polytopes (up to 1000 inference runs)
- Small polytopes: FRST space quickly exhausted; large polytopes: continued exploration
- Flat rates when all (including duplicates) are counted due to independence of guesses

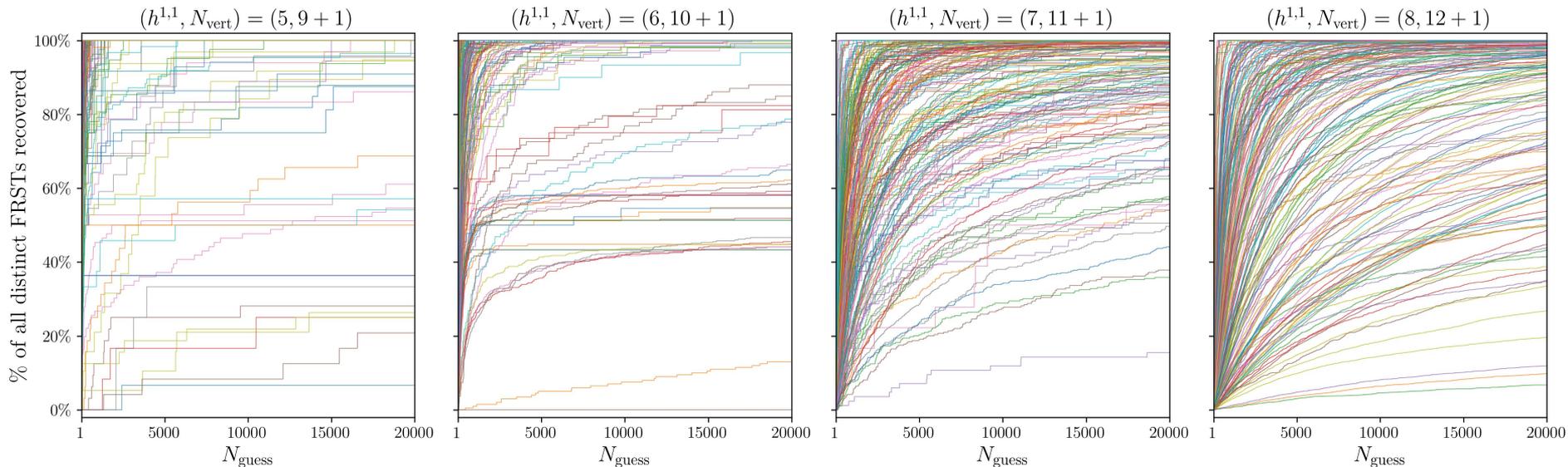
CYTransformer - Height vector similarity score histogram

$$(h^{1,1}, N_{\text{vert}}) = (8, 12 + 1)$$



- Distribution of similarity scores of distinct FRSTs from 20,000 guesses per test polytope
- CYTransformer reproduces the overall shape of the population distribution even with partial recovery - **unbiased sampling**

CYTransformer - Recovery curves (per polytope)



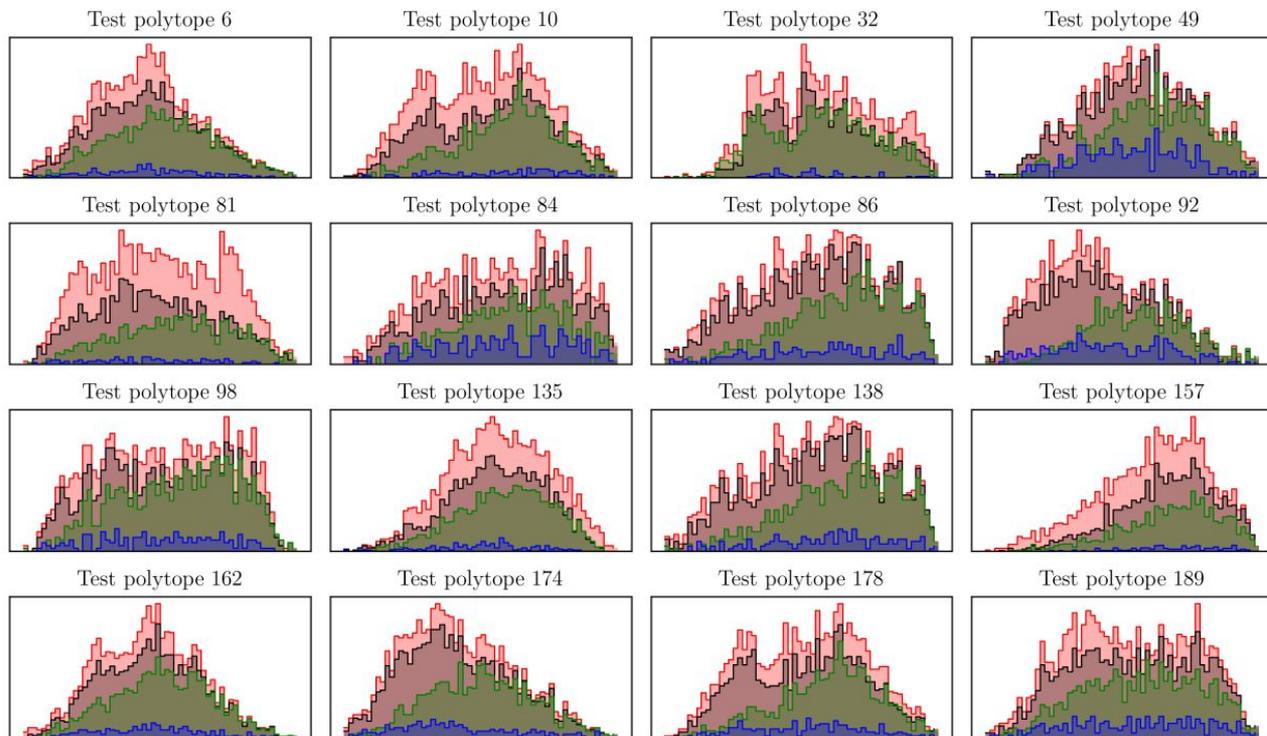
- **High recovery for most test polytopes shows that CYTransformer generalizes well**
- Some curves plateauing early shows that some polytope geometries are difficult to learn
 - Possible failure modes
- Slow-rising curves might just be due to very large FRST spaces

The hybrid approach - Transformer-seeded fast sampler

- Best of both worlds: CYTransformer + fast sampler
 - Global explorer + local scanner
 - Unbiased global coverage + high local efficiency
- Algorithm
 - First N guesses
 - CYTransformer generates a pool of representative FRST seeds
 - For each subsequent guess
 - Randomly pick a FRST seed
 - Apply the fast sampler: Perturb around the seed

Transformer-seeded fast sampler - Height vector similarity score histogram

$$(h^{1,1}, N_{\text{vert}}) = (8, 12 + 1)$$



Red - Population

Blue - Seeds at $N = 500$

Black - Hybrid final result

Green - Pure fast sampler

- (Not shown) Hybrid recovers more FRSTs than CYTransformer or fast sampler alone
- With just 500 CYTransformer runs for seeds, hybrid recovers the population shape
- **Hybrid maintains unbiasedness**

