Generating Calabi-Yau Manifolds with Machine Learning

Elli Heyes
City, University of London

Collaborators:

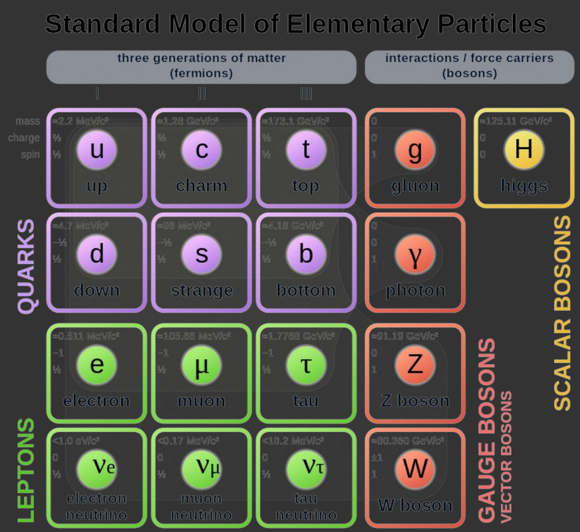
Per Berglund, Giorgi Butbaia, Yang-Hui He, Edward Hirst
Vishnu Jejjala, Andre Lukas

Outline

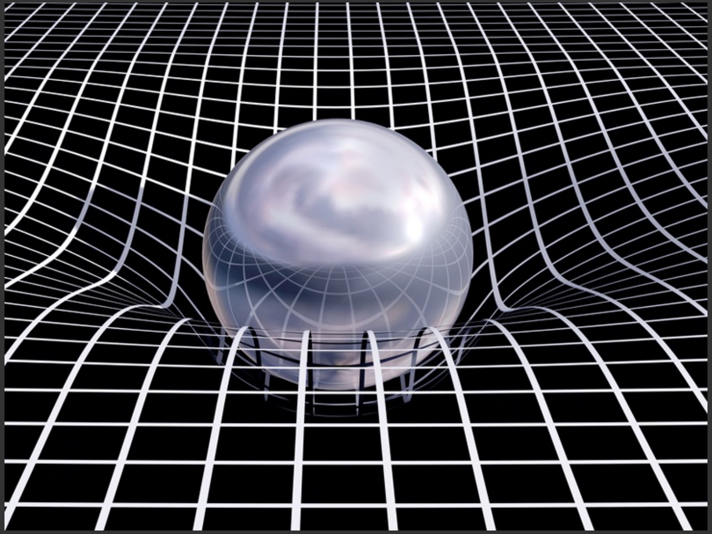
1. Motivation
   - introduction to string theory
   - Calabi-Yau manifolds in string theory
2. Background
   - toric construction of a Calabi-Yau manifold
3. Generating polytopes
   - genetic algorithms
   - results
4. Generating triangulations
   - reinforcement learning
   - results
5. Future Directions

① Motivation

Standard Model



General Relativity

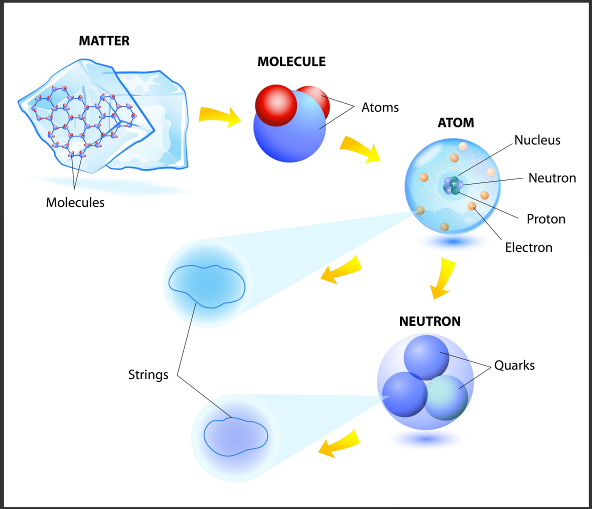


① Motivation

String Theory

In string theory 0-dimensional particles are replaced by a 1-dimensional string

Different vibrational modes of the string give us different particles in the standard model



1 Motivation

String Theory

Problem: String theory only works in 10-dimensions of spacetime, but we experience only 4.

Solution: Hide the extra dimensions where nobody can see them.

$$M_{10} = \mathbb{R}^{1,3} \times M_6 \leftarrow \text{small \& compact}$$

M_6 must be a Calabi-Yau manifold

A Ricci-flat Kähler manifold with holonomy group $SU(3)$ is called a Calabi-Yau manifold.

① Motivation

String Theory

Problem: The landscape is too big

Solution: Use machine learning to identify "good" regions of the landscape

Problem: No analytical Ricci-flat metric

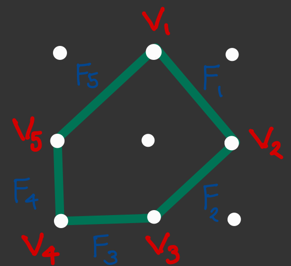
Solution: Use machine-learning to engineer aproximations

② Background

Calabi-Yau Manifolds

Toric Varieties X_Δ can be built from polytopes Δ

M



$\Longrightarrow$

N

$\Delta = \{\sum c_i v_i \in M_{\mathbb{R}} \mid c_i \in \mathbb{R}, \sum c_i = 1, c_i \geq 0\}$ vertex

$v_i \in M_{\mathbb{R}}$

$= \{m \in M_{\mathbb{R}} \mid \langle u_j, m\rangle + a_j \geq 0, \forall j\}$ hyperplane

$u_j \in N_{\mathbb{R}} \quad a_j \in \mathbb{R}$

$\Sigma_\Delta = \{\sigma_j\}$

$\sigma_j = \mathrm{Cone}(u_j)$

$\Longrightarrow$

X_Δ

$X_j = \mathrm{Specm}(\mathbb{C}[\sigma_j^\vee \cap M])$

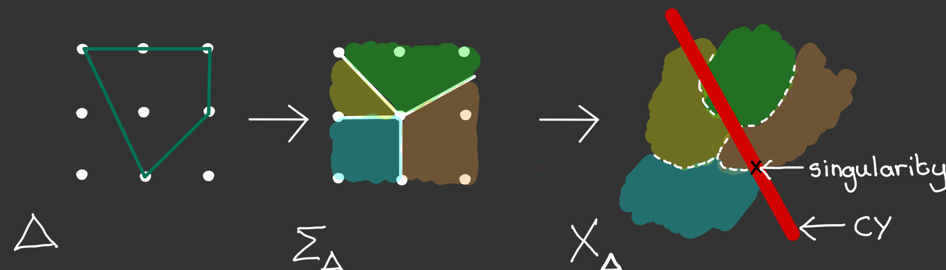
② Background

Calabi-Yau Manifolds

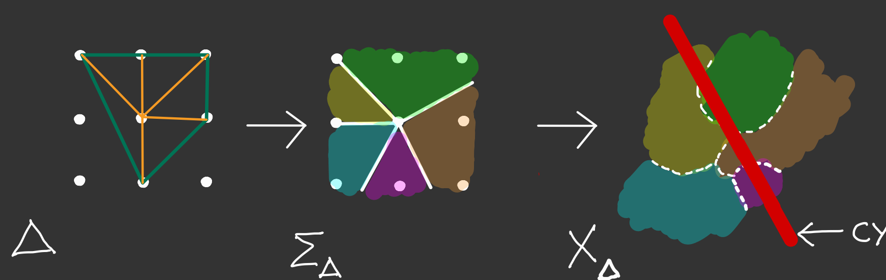
If Δ is reflexive then:

i) X_{Σ_Δ} is a Fano variety with canonical singularities

ii) any generic anticanonical hypersurface in X_{Σ_Δ} is a Calabi-Yau variety

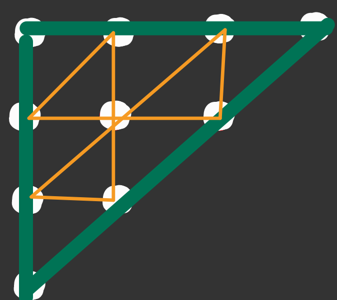


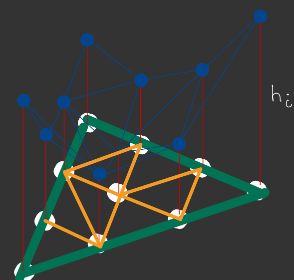
Desingularisations of X_Δ are defined by Fine Regular Star Triangulations (FRSTs) of Δ

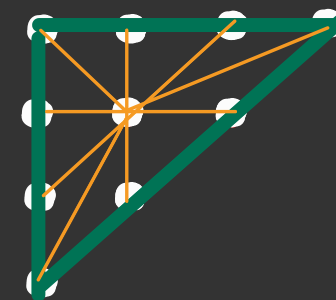


② Background

FRSTs



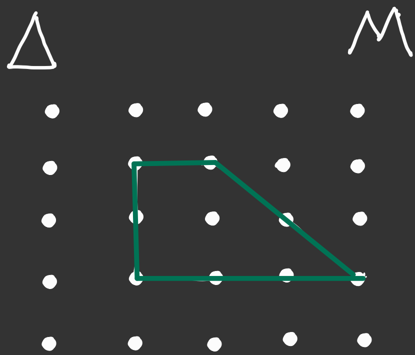


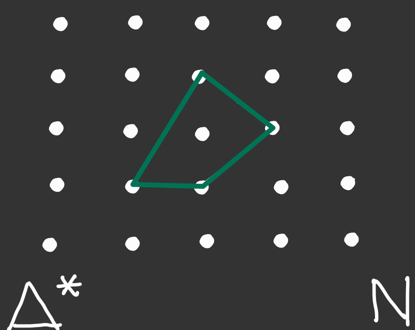


FINE	REGULAR	STAR
Every point is included	Can be obtained by assigning a height to every point, raising the polytope up and projecting down the faces	The origin is a vertex of every simplex
Ensures all singularities are resolved	Ensures toric variety is Kähler	Ensures we can produce a fan

② Background

Reflexive Polytopes





$\Delta = \{\sum c_i v_i \in M_{\mathbb{R}} \mid c_i \in \mathbb{R}, \sum c_i = 1, c_i \geq 0\}$ vertex
$\quad v_i \in M_{\mathbb{R}}$

$= \{m \in M_{\mathbb{R}} \mid \langle u_j, m\rangle + a_j \geq 0, \forall j\}$ hyperplane
$\quad u_j \in N_{\mathbb{R}} \quad a_j \in \mathbb{R}$

$\Delta^* = \{n \in N_{\mathbb{R}} \mid \langle n, m\rangle \geq -1, \forall m \in \Delta\}$ dual

Lattice : $v_i \in M \quad \forall i$

IP: $\ell^*(\Delta) = \{0\}$

Defintion : Δ is called reflexive if

i) Δ & Δ^* are lattice	i) Δ satisfies IP
ii) Δ & Δ^* satisfy IP	ii) $a_i = 1 \quad \forall i$

② Background

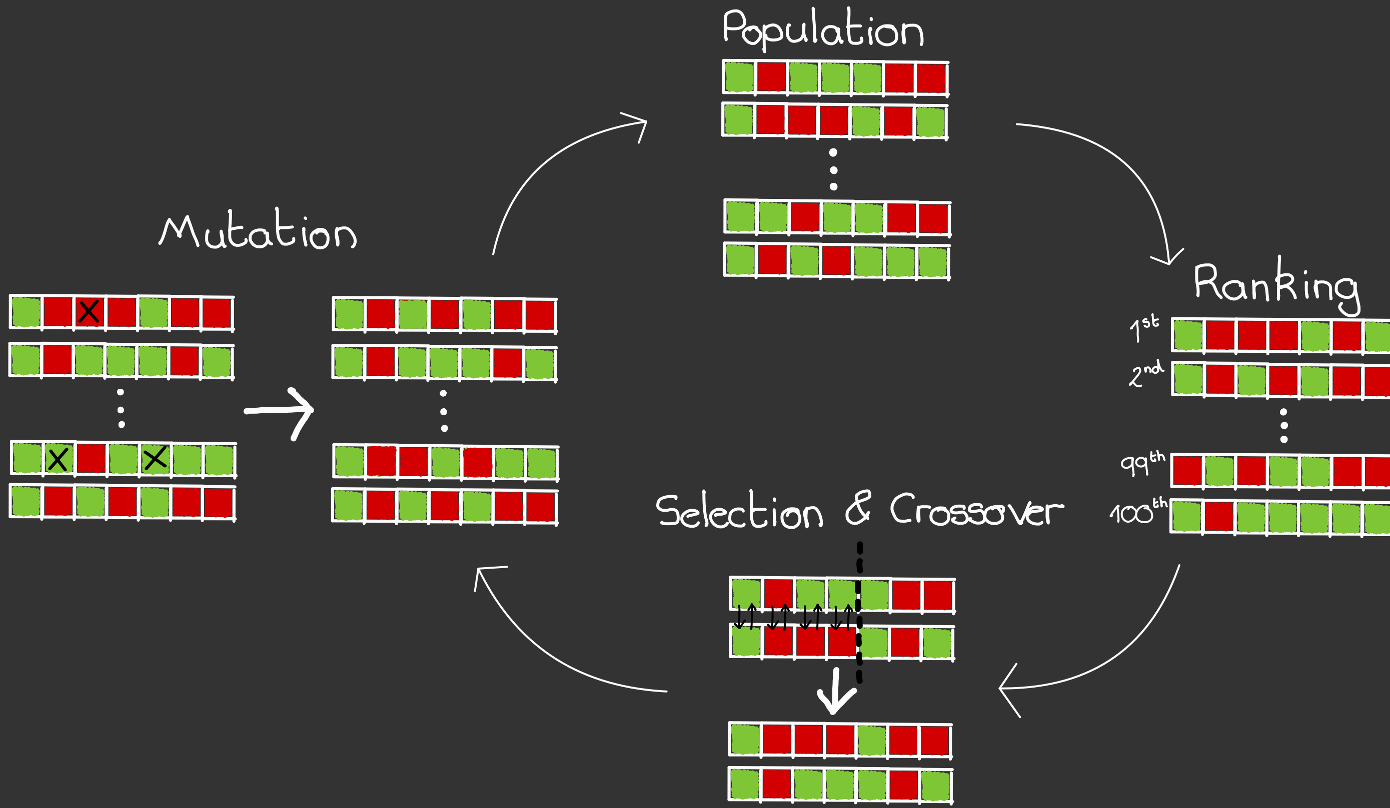
Reflexive Polytopes

Classification

dimension	# reflexive	Calabi-Yau	
2	16	elliptic curves	
3	4319	K3	(Kreuzer & Skarke)
4	473,800,776	CY 3-folds	(Kreuzer & Skarke)
5	>185,269,499,015	CY 4-folds	(Skarke & Schöller)

③ Polytopes

Genetic Algorithms



③ Polytopes

Fitness Function

$$\Delta = \{ m \in M_{\mathbb{R}} \mid \langle u_j, m \rangle + a_j \geq 0, \forall j \} \quad u_j \in N_{\mathbb{R}} \quad a_j \in \mathbb{R}$$

$$\boxed{f(\Delta) = \omega_1 (IP(\Delta) - 1) - \frac{\omega_2}{R} \sum_{i=1}^{R} |a_i(\Delta) - 1| - \omega_3 |N_p(\Delta) - N_{p,0}|}$$

- $IP(\Delta) = \begin{cases} 1, & \text{if } \Delta \text{ satisfies } IP \\ 0, & \text{otherwise} \end{cases}$
- $N_p(\Delta) = \#$ points of Δ and $N_{p,0} =$ desired $\#$ points
- $\omega_1, \omega_2, \omega_3 \in \mathbb{R}^{\geq 0}$ are weights

3 Polytopes

Method

1 GA "run"

1. Generate a random population P_0 of size N
2. Evolve P_0 over M generations

$$P_0 \to P_1 \to \dots \to P_{M-1} \to P_M$$

3. Extract any reflexive polytopes from $\{P_0, \dots, P_M\}$

4. Repeat steps 1-3 untill all reflexive polytopes are found

③ Polytopes

Results

2D

Mutation rate : 0.5%
generations : M = 500
Population size : N = 200
Max # vertices : 6
Vertex coordinate range : [-4,4]

- # unique reflexive polytopes : 16
- Size of environment : $\sim 10^{11}$
- GA found all unique reflexive polytopes in 1 run !

3D

Mutation rate : 0.5%
generations : M = 500
Population size : N = 450
Max # vertices : 14
Vertex coordinate range : [-8,8]

- # unique reflexive polytopes : 4319
- Size of environment : $\sim 10^{51}$
- GA found all unique reflexive polytopes in 117251 runs !

③ Polytopes

Results

4 D

mutation rate: 0.5%   vertex coordinate range: [-4, 4]
\# generations: M = 500   max # vertices = # points - 1

# points	# states	pop. size	# refl. poly	# GA runs
6	$\sim 10^{19}$	400	3	5
7	$\sim 10^{22}$	300	25	30
8	$\sim 10^{26}$	400	168	60
9	$\sim 10^{29}$	300	892	9378
10	$\sim 10^{33}$	350	3838	9593

③ Polytopes

Results

5 D

mutation rate: 0.5%
generations: M = 500
Vertex coordinate range: [-4, 4]
max # vertices = # points - 1

# points	# states	pop. size	# refl. poly.	# GA runs
7	$\sim 10^{28}$	350	9	36
8	$\sim 10^{32}$	350	115	1278
9	$\sim 10^{37}$	450	1385	7520
10	$\sim 10^{41}$	750	12661	31857
11	$\sim 10^{46}$	650	94556	376757

③ Polytopes

Results

Reflexive polytopes give rise to families of CYs

CY 4-folds with different Hodge numbers $h^{1,1}, h^{1,2}, h^{1,3}, h^{2,2}$ are inequivalent

$$h^{1,1} = \ell(\Delta^*) - 6 + \sum_{\mathrm{codim}\,\theta^* = 1} \ell^*(\theta^*) - \sum_{\mathrm{codim}\,\theta = 1} \ell^*(\theta^*) \cdot \ell^*(\theta)$$

$$h^{1,2} = \dots \,, \quad h^{1,3} = \dots \,, \quad h^{2,2} = \dots$$

We found new CY 4-folds with new $h^{i,j}$

③ Polytopes Targeted Search

Unbroken $\mathcal{N}=1$ SUSY for 11d SUGRA on CY 4-fold requires

$$\chi \% 24 = \chi \% 224 = \chi \% 504 = 0$$

Fitness Function

$$f(\Delta) = \omega_1 (IP(\Delta) - 1) - \frac{\omega_2}{k} \sum_{i=1}^{k} |a_i(\Delta) - 1| - \omega_3 \sum_{\delta \in \{24,224,504\}} \chi(\Delta) \bmod \delta$$

GA finds examples after just a few runs

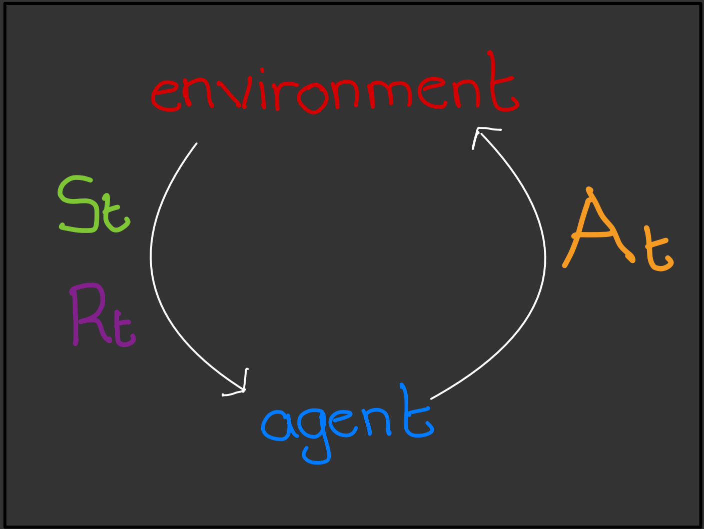
Generative machine learning methods can generate Calabi-Yau manifolds of a certain type

5 Triangulations

Reinforcement Learning

In reinforcement learning an agent interacts with its environment in time steps t.

- At each t the agent receives the current state S_t and reward R_t
- It chooses an action A_t which is then sent to the environment
- The environment moves to a new state $S_{t+1} = A_t(S_t)$
- The goal of the agent is to learn a policy $\pi : S \times A \to [0,1]$, $\pi(s,a) = Pr(A_t = a \mid S_t = s)$ that maximises the expected cumulative reward

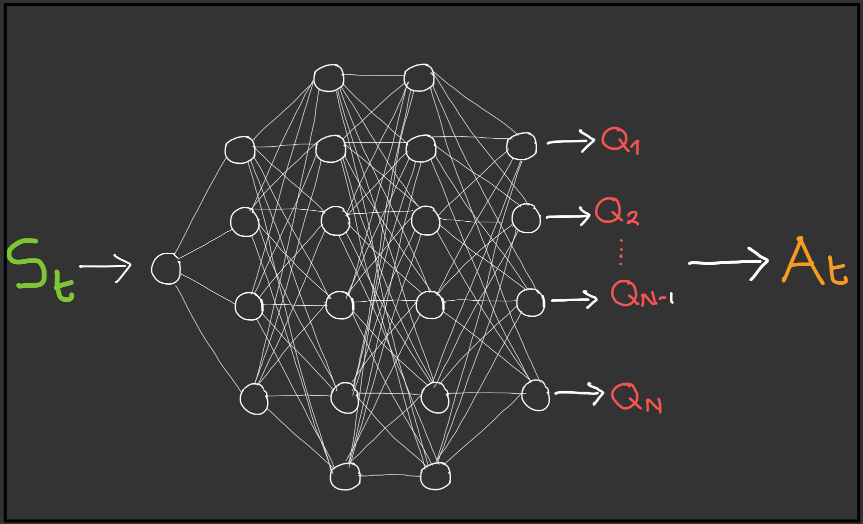


⑤ Triangulations

Deep Q-Learning

Q-learning is based on a value matrix Q that assigns quality of a given action A_t when the environment is in a given state S_t

In deep Q-learning a neural network is used to represent Q.



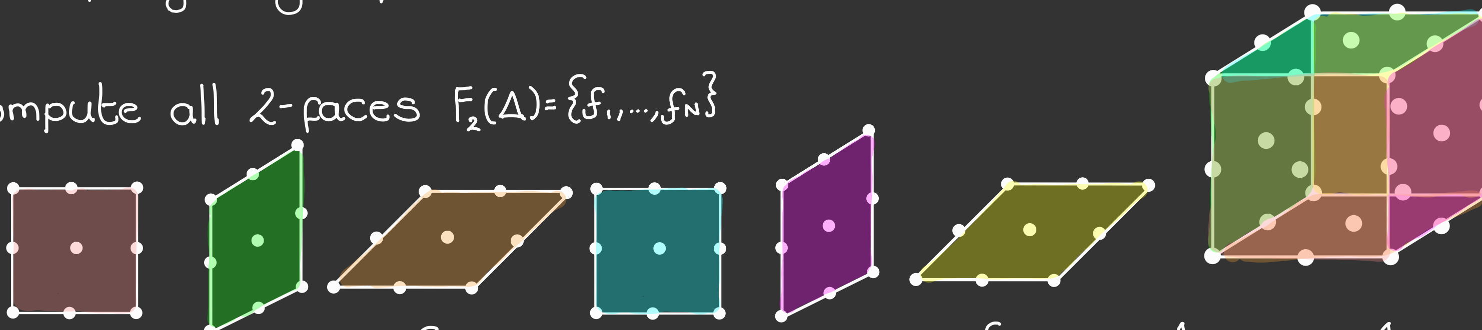
Training:

1. Randomly generate state S_0
2. Either i) randomly pick an action A_t or
   ii) pick action A_t with largest Q value from NN output
3. Get new state S_{t+1}
4. Compute Q-value $Q(S_t, A_t) = R(S_{t+1}) - R(S_t)$
5. Repeat 2-4 until terminated
6. Train on $\{(S_0, Q(S_0, A_0)), \dots, (S_N, Q(S_N, A_N))\}$
7. Repeat 1-6 X times

5 Triangulations — State Space

Any 2 FRSTs T_1, T_2 of a polytope Δ with the same 2-face restriction are topologically equivalent

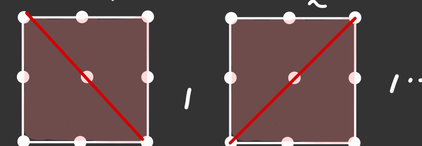
1. Compute all 2-faces $F_2(\Delta)=\{f_1,\dots,f_N\}$



f_1 f_2 f_3 f_4 f_5 f_6

2. For each f_i compute all fine triangulations $\{T_1^i,\dots,T_{M_i}^i\}$

T_1^1 , T_2^1 , ...



3. A triangulation state of Δ is given by picking a T_j^i for each f_i

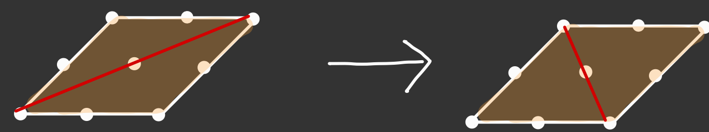
$$\text{e.g. } T=\{\{1,0,..,0\},\{0,\dots,1,\dots,0\},\dots,\{0,1\}\}$$

5 Triangulations

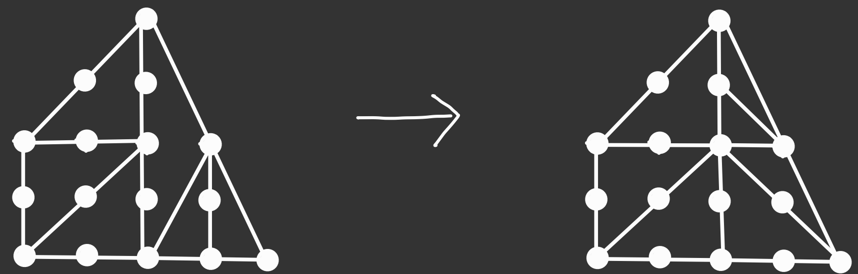
Action Space

The actions consist of swaping T_j^i for T_k^i for all i, j, k

e.g. $\{\{1,0,..,0\},\{0,...,1,...,0\},... ,\{0,1\}\} \longrightarrow \{\{1,0,..,0\},\{0,...0,1\},... ,\{0,1\}\}$



Note: Actions don't always correspond to bistellar flips



5 Triangulations

Reward

$$R(S,A) = F(S_{t+1}) - F(S_t)$$

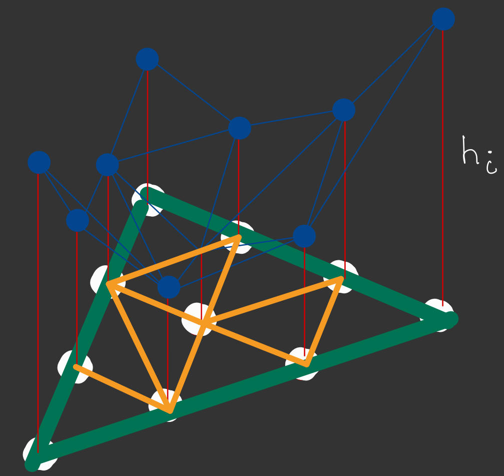
where $F: S \longrightarrow [0,1]$ is a fitness function

- [x] All T_j^i are fine so combined triangulation is always fine
- [x] We can always make a triangulation star
- [ ] All that remains is to check regularity of combined triangulation

Secondary Cone

If $h=(h_1,\dots,h_n)$ generates a triangulation T, then ch for any $c\in\mathbb{R}$, $c>0$ also generates T

The collection of all heights h that define T form the interior of a cone called the secondary cone C.



T is regular if and only if C is full dimensional.

5 Triangulations

Fitness

To check regularity of combined triangulation:

1. Compute the secondary cone for each 2-face triangulation

$$C_i, \quad i=1,\dots,N$$

2. Compute the intersection cone

$$C = C_1 \cap \dots \cap C_N$$

3. If C is full dimensional then the combined triangulation of S is a regular triangulation

$$\boxed{F(S) = \begin{cases} 1 & \text{if } C \text{ is full-dimensional} \\ 0 & \text{otherwise} \end{cases}}$$

5 Triangulations

Results (so-far)

- Generated triangulations for 4d reflexive polytopes with low $h^{1,1}$

Next Steps

- Generate triangulations for $h^{1,1} = 491$ case
- Add CY constraints into fitness
- Combine with polytope generation algorithm

5 Future Directions

string phenomology constraints ⟹ generative machine-learning model ⟹ example string models

⇓

geometric constraints

Thank You

arXiv

GitHub