

**CAYLEYPY-4: AI-HOLOGRAPHY. TOWARDS ANALOGS OF HOLOGRAPHIC
STRING DUALITIES FOR AI TASKS.**

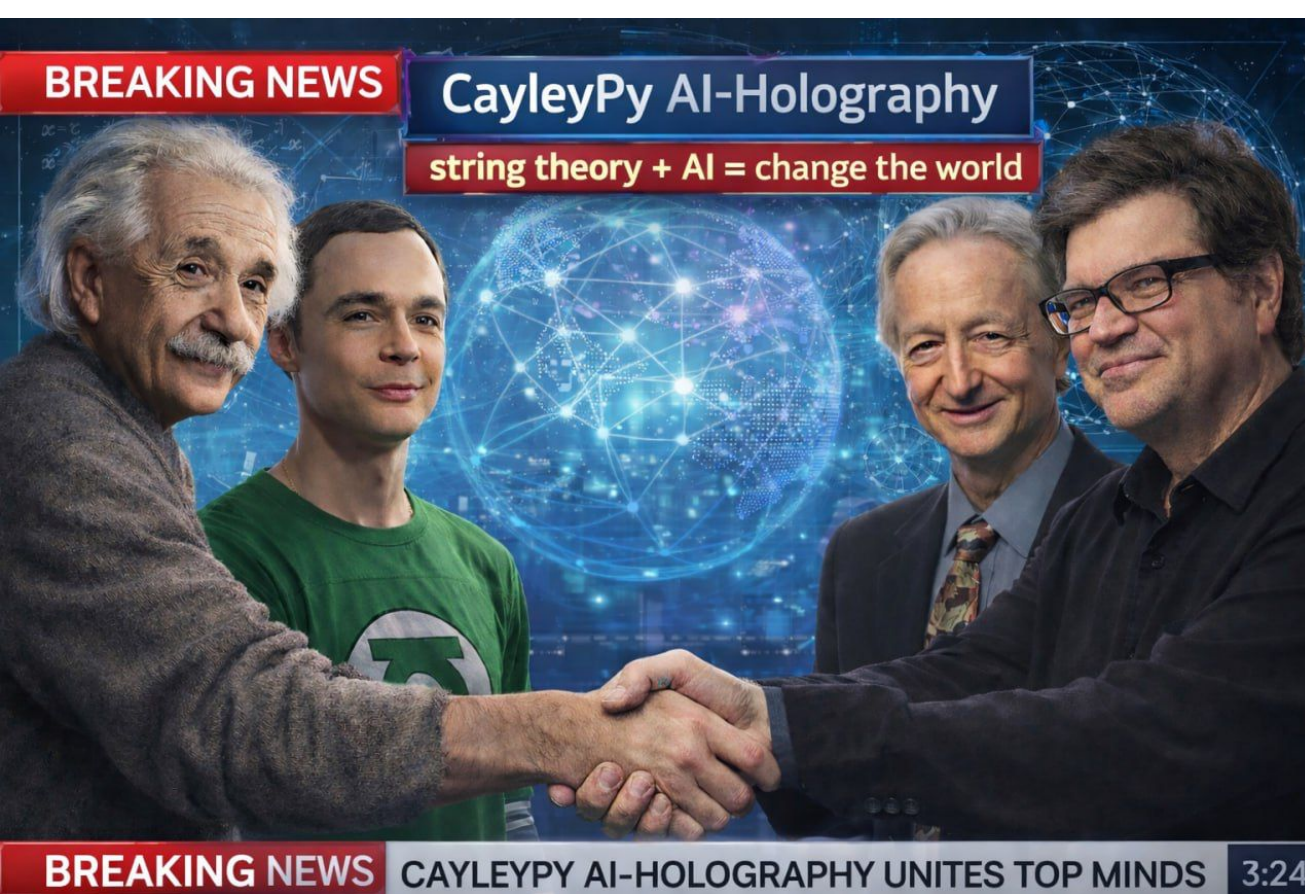
(PRELIMINARY VERSION 0.053 NOT FINISHED, NOT FOR DISTRIBUTION)

A. CHERVOV, F. LEVKOVICH-MASLYUK, A. SMOLENSKY, F. KHAFIZOV, I. KISELEV, D. MELNIKOV,
S. KRYMSKII, S. KUDASHEV, D. SHILTSOV, I. KOLTSOV, M. OBOZOV, V. KIROVA, E.V. KONSTANTINOVA,
A. SOIBELMAN, S. GALKIN, L. GRUNWALD, A. KOTOV, A. ALEXANDROV, S. LYTKIN, D. FEDORIAKA,
A. CHEVYCHELOV, Z. KOGAN, A. NATYROVA, L. CHELDIEVA, O. NIKITINA, S. FIRONOV,
A. VAKHRUSHEV, A. LUKYANENKO, V. ILIN, D. GORODKOV, N. BOGACHEV, I. GAIUR, M. ZAITSEV,
F. PETROV, L. PETROV, T. GAINITSEVA, A. GAVRILOVA, M. N. SMIRNOV, N. KALININ, A. KHAN, K. JUNG,
H. MOUSSET, H. ISAMBERT, AND O. DEBEAUPUIS

Alexander Chervov, Febraury 2026

CayleyPy project - crowdsourcing - welcome to join if you have "couple" free hours per week

- CayleyPy-1 (Cube) - first ML approach to find paths on graphs up to 10^{75} (e.g. Rubik's cube 555, beating specialized solvers for cubes 444,555), NIPS-2025 Spotlight
- CayleyPy-2 (RL) - applications to real mathematical research problems - estimations of diameters of certain Cayley graphs which are open for 50+ years. And update of the methodology combining it with RL. Adv.MathPhys.
- CayleyPy-3 (Growth) - hundreds new mathematical conjectures and first official release of CayleyPy open source library. Oral talk NIPS subconf
- CayleyPy-4 (AI-holography) - string dualities -> AI , multiple new mathematical conjectures
- 400+ stars on Github; Documentation; Tutorials; 500+ notebooks with experiments; Kaggle challenges; 100+ community
- <https://cayleypy.github.io/cayleypy-docs/index.html>



String duality:
difficult problems ->
more tractable

(“strong coupling” ->
“weak coupling”)

**The most powerful
technique** ever
created by
generations of
geniuses from
Einstein to “Sheldon
Cooper”

Finding: towards a
way to apply it to AI
tasks

Quiz: what physics Sheldon Cooper was doing ? (“Big bang theory”)

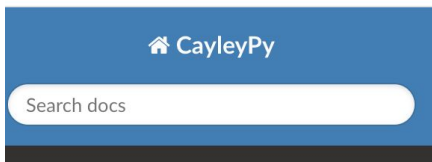
Duality key example: Holographic AdS/CFT correspondence - “the most
highly cited article in the field of high energy physics”

CayleyPy: AI-based open-source Python library to handle GOOGOL size graphs

150+ stars on Github; Documentation; Tutorials; 300 notebooks with experiments

<https://cayleypy.github.io/cayleypy-docs/index.html>

← → ↺ 🌐 cayleypy.github.io/cayleypy-docs/generated/cayleypy.Permutatio



CayleyPy API Reference

Core classes and functions

Graphs library

cayleypy.PermutationGroups

PermutationGroups

cayleypy.MatrixGroups

cayleypy.Puzzles

cayleypy.GapPuzzles

cayleypy.create_graph

cayleypy.prepare_graph

Beam search and ML

Special algorithms

`all_cycles (n)`

`all_transpositions (n)`

`block_interchange (n)`

`burnt_pancake (n)`

`conjugacy_classes (n, classes)`

`consecutive_k_cycles (n, k)`

`coxeter (n)`

`cubic_pancake (n, subset)`

`cyclic_coxeter (n)`

`derangements (n)`

`full_reversals (n)`

`generalized_stars (n[, k])`

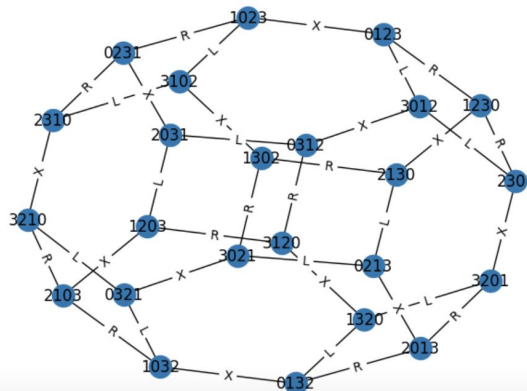
`increasing_k_cycles (n, k)`

```
import networkx as nx
from cayleypy import CayleyGraph, PermutationGroups

graph = CayleyGraph(PermutationGroups.lrx(4))
result = graph.bfs( return_all_edges=True, return_all_hashes=True)
G = result.to_networkx_graph()

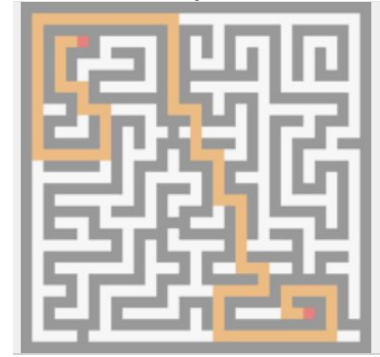
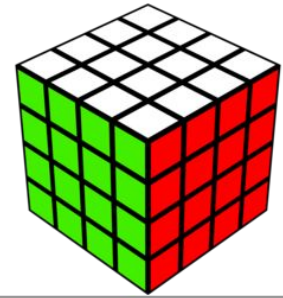
pos = nx.kamada_kawai_layout(G)
nx.draw(G, pos, with_labels=True)
edge_labels = nx.get_edge_attributes(G, 'label')
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels)

None
```



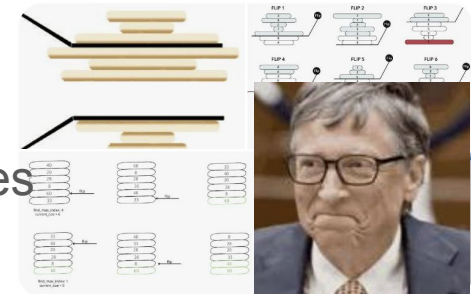
First task - FOUR equivalent ways to explain:

1. (Fun) Solve a Rubik's cube (any puzzle)
2. (Math) Decompose group element into product of generators, e.g. decompose a matrix into product of given matrices
3. (Graph) Find a path on a graph (Cayley graph)
4. (CS) Sorting problem



Pancake sorting

Sorting algorithm :



Terminology: **Diameter** of graph - largest distance between nodes
= God's number = worst case performance of God's algorithm

Results and SOTA

1. C. Sims 1970: Schreier–Sims algorithm (permutation groups)
2. Knuth, Donald E. (1991): randomized version (10^{20} - 10^{30})
3. **CayleyPy beats by orders and orders of magnitude (10^{50} - 10^{120})**

Complexity:

1. 1981 NP-hard: optimal solutions and diameters
2. 1998-....: optimal solution
NP-complete for Rubik's cube, etc

n	GAP Length	Ideal Length $n(n-1)/2$	AI Length
9	41	36	36
10	51	45	45
11	65	55	55
12	78	66	66
13	99	78	78
14	111	91	91
15	268	105	105
16	2454	120	120
17	380	136	136
18	20441	153	153
19	3187	171	171
20	217944	190	190
21	-	210	210

Testbed for: automatic provers, "chatGPT", robotics ...

Tasks - points for colab - AI-part - improve + bench

1. ML/RL

Develop (near) optimal solver for groups/graphs of sizes up to 10^{40} - 10^{50} . (In particular 444 Rubiks cube and send paper to Nature - first ever optimal solution for the 40 years old problem).

Develop any kind of solvers up to 10^{200} - beat Go size: 10^{170}

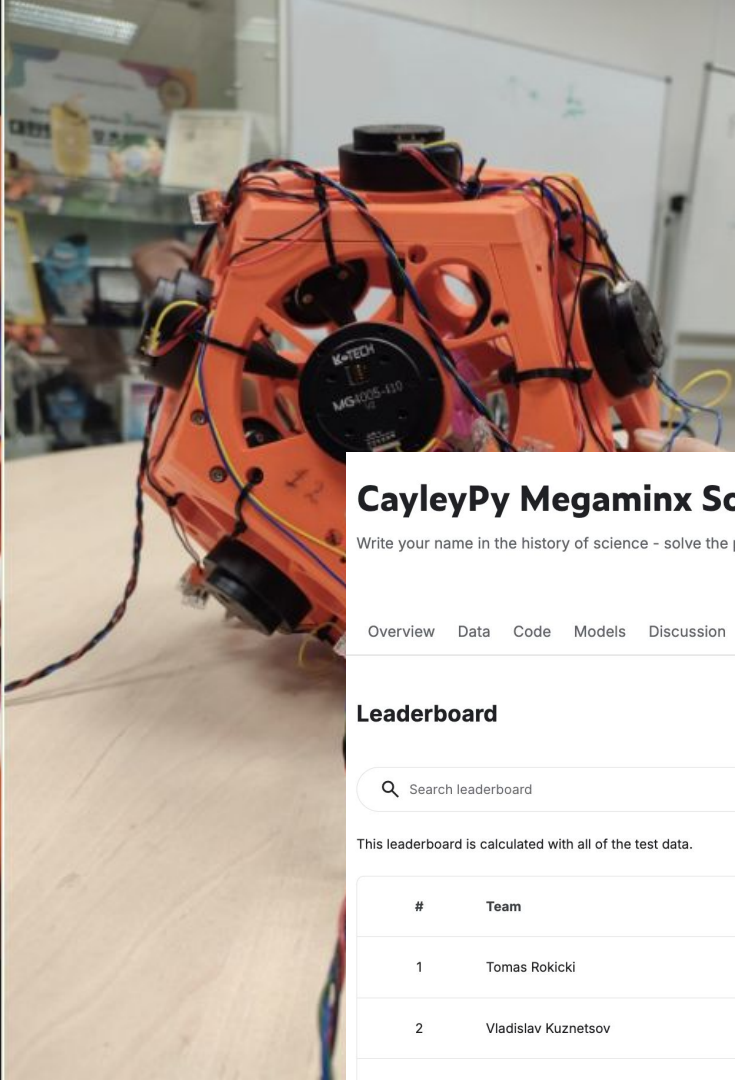
Key issue: no scaling - more training does not improve performance

How to combine algorithms with ML ?

2. LLM:

LLM to invent/improve existing algorithms - achieve optimal - where it is not NP-hard (e.g. LRX)

3. Many further tasks - try advanced RL techniques PPO, DPO, try advanced models. Use our task to benchmark RL and NN. Etc.



Megaminx Challenge

CayleyPy Megaminx Solve Optimally

Write your name in the history of science - solve the problem unsolved from 1982



[Overview](#) [Data](#) [Code](#) [Models](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [Submissions](#) [Settings](#)

Leaderboard

[Raw Data](#)

Search leaderboard

This leaderboard is calculated with all of the test data.

#	Team	Members	Score	Entries	Last
1	Tomas Rokicki		93606	2	3mo
2	Vladislav Kuznetsov		109478	3	23d

Tasks - points for colab - math part

1. Diameters of various graphs.

E.g. Pancake - we have a conjecture - need good ML model to check it. (Classical problem)

2. Various other things on graphs - basically all characteristics we can make discoveries

3. Big thing - "string duality for AI"

String duality for graphs and AI

Explain first - quasi polynomiality conjecture

Explain how it is related to polygons

Surprise: Diameters - "findable"(?!) = quasi-polynomials

Example of quasi polynomial: $n^2/3$ if $n \equiv 0 \pmod 3$, $(n^2-1)/3$ if $n \equiv 1, -1 \pmod 3$

1. Conjecture 1: Diameters of S_n are quasi-polynomials (for constructive generators)
2. Conjecture 2: Length (word metric) of ANY element is quasi-polynomial (constructive element and constructive generators)

Example (<https://oeis.org/A039745>):

Two generators L - left cyclic shift, $X = (1,2)$

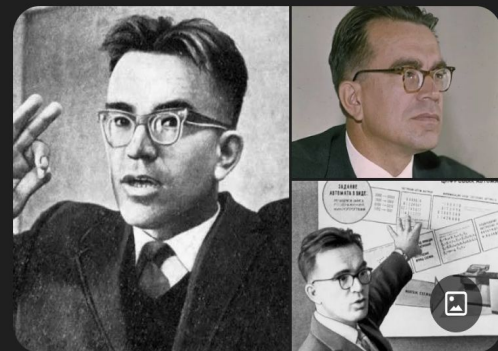
"father of Soviet cybernetics" V.M.Glushkov 1968 problem:

New conjecture: $n > 4$: diameter $(3n^2 - 8n + 9)/4$, n odd

$(3n^2 - 8n + 12)/4$, n even . Checked $n < 16$

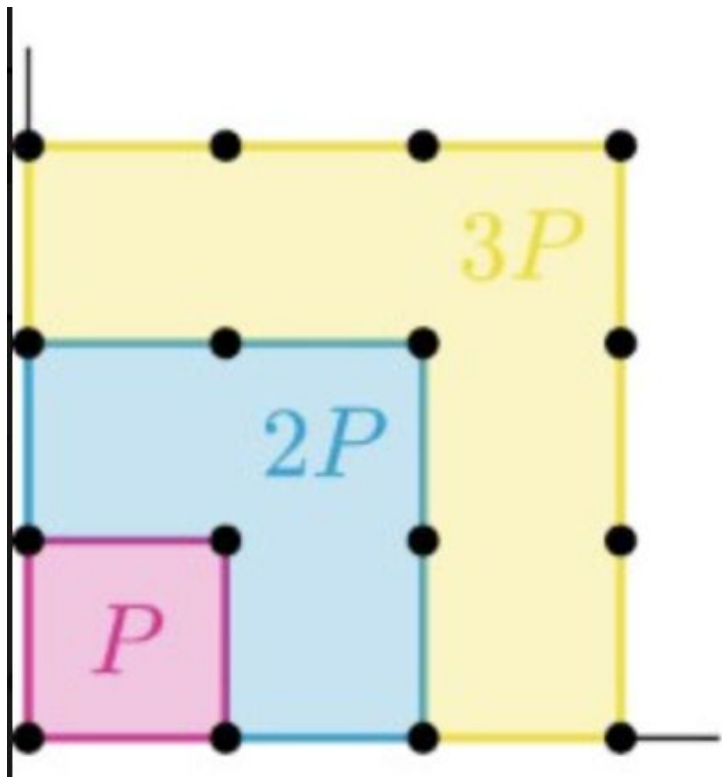
Victor Glushkov

Soviet computer scientist



Victor Mikhailovich Glushkov was a Soviet computer scientist. He is considered to be the founding father of information technology in the Soviet Union and one of the founding fathers Soviet cybernetics.

Ehrhart polynomial - main source of quasi-polynomials



Take a rational polygon (or polytope) count integer points in it and in its n -scaled copy - theorem of Ehrhart says values will be quasi-polynomials.

The only way to explain conjecture that diameters and word-metrics are quasi-polynomials - is to propose that there exist planar polygon and holography such that diameters/words metrics = numbers of integer points in/under the polygon/string

Complexity = Area (same as in AdS/CFT)

Graph ↔ Rectangle Holography

"Particle on a graph dual to a string on the rectangle" – click nodes or paths to explore the holography map

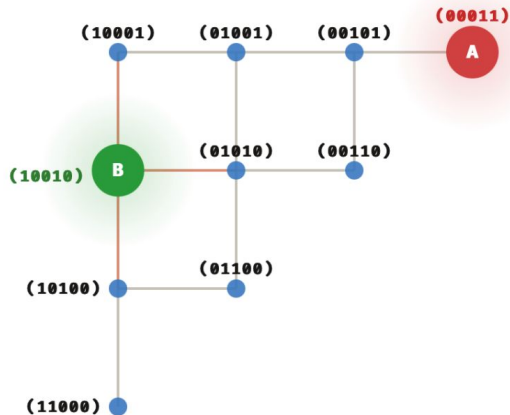
N = 5 K = 2 Gr(2, 5) over F₃ – 10 nodes

HELP

Schreier Graph (CFT-Side)

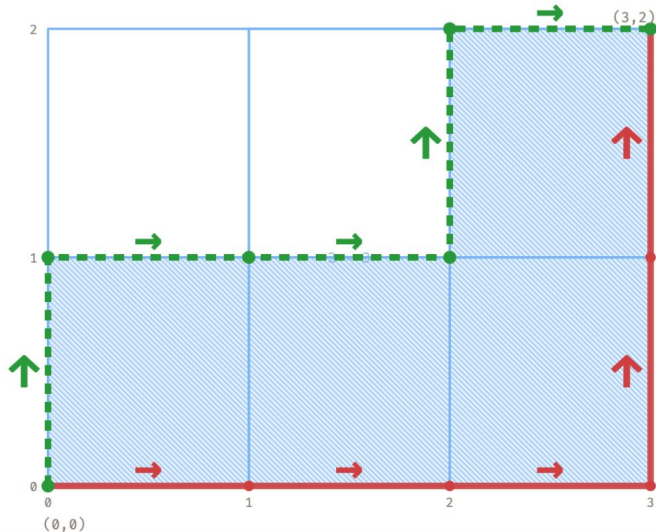
First click: node A. Second click: node B.

CLEAR SELECTION



Rectangle Paths (ADS-Side)

☐ Highlight area under path

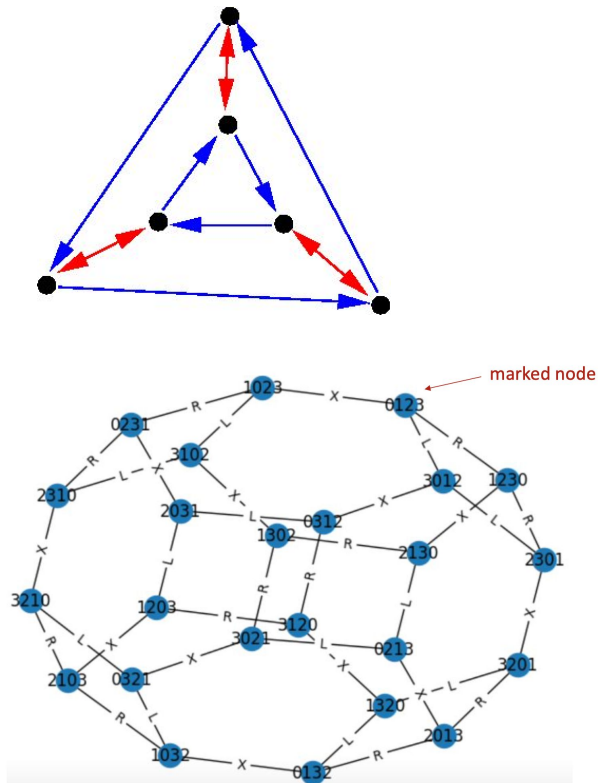


For example -
difficult task -
compute length
of the shortest
path
("complexity") -
converts to
compute area
under the curve
- which is much
more tractable

"Cayley like" graphs - need labels (tokens) on EDGES

Basic idea:

- Fix some set of matrices (or transformations) M_i (edges)
- Fix some set of vectors (or states) v_i (nodes)
- Connect two vectors v and w if there exists M such that $v = M w$



Paths on such graphs - sentences in alphabet M_i :
E.g. Red.Blue.Red.Blue...
Or L.R.X.R.X.L.R.L.X...

What are the languages ? Texts = paths on such graphs
Physically - particle trajectories on such graph

Natural:

Hello world...

Reinforcement learning e.g. chess:
e2-e4; e7-e5;...

SMILES (molecules)

CC1CCC/C(C)=C1/C=C/C(C)=C/C...

Proteins:

MQYLNIEDCNAMAFCAKMR...

Any language can be defined by set
of all texts in it - all sequences

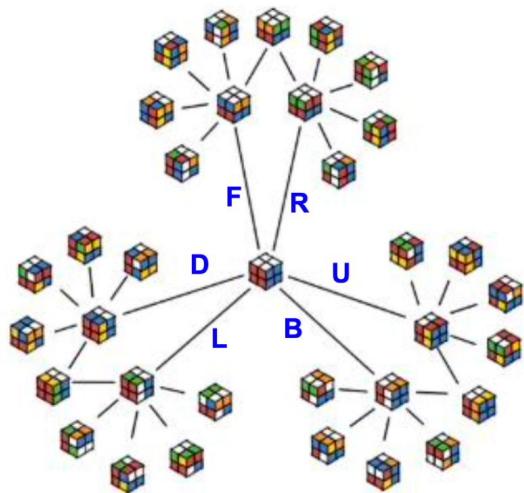
- "Geodesic language of graph"
- only shortest paths
- Almost-geodesic - not longer than say 10 than shortest
- So on - having edge labeled graph and imposing various conditions on graphs you can define various languages
- Physically - different particle dynamics on graphs - defines different trajectories = different languages

Example of Rubik's cube. "God's number" (diameter)

How to solve
a Rubik's cube?



$\sim 10^{20}$ configurations, 6 allowed moves F, R, U, B, L, D



Cayley graph

nodes = configurations

edges = moves

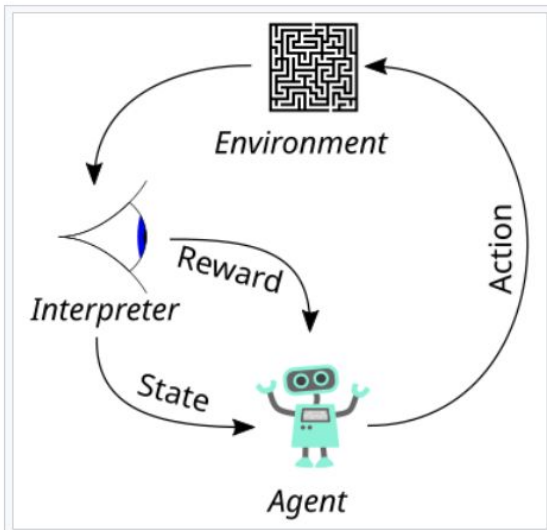
Our task: solve cube = find path from given node to starting point

Very similar to games (chess/Go),
robotics, language, etc

"God's number" = diameter - how many moves you need to solve the hardest case (assuming you have perfect (God's) algorithm)

Tomas Rokicki [et.al.](#)
(2010): 20/26

NP-hard in general



The typical framing of a reinforcement learning (RL) scenario: an agent takes actions in an environment, which is interpreted into a reward and a state representation, which are fed back to the agent.

Reinforcement learning = = Explained in 1 sec via graphs

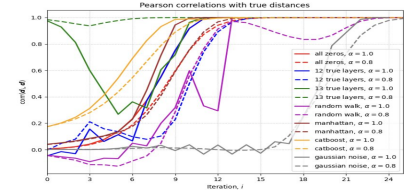
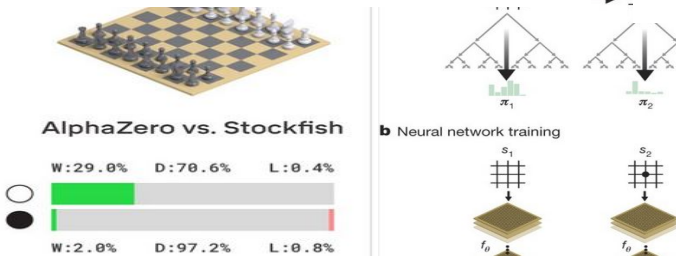
Vocabulary:

States = Nodes of the graph
 Actions = Edges
 Reward(Penalties) = Weights of edges
 Cumulative reward = Length of the path

$$d(g) = 1 + \min_{n: \text{neighbours of } g} d(n) \text{ Bellman equation}$$

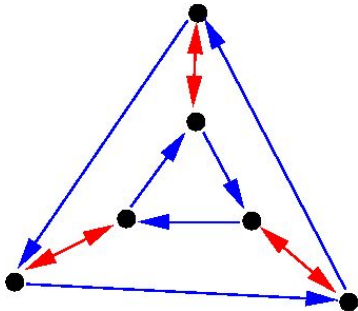
$$d(e) = 0 \text{ boundary condition}$$

Solving Bellman = simple iteration



Key tasks in ML = predict particle trajectory given initial or boundary conditions.

- GPT models (given prompt) you need to continue the text - that is like given initial condition - you need to continue trajectory of a particle on a graph
- "Win the game/reach the goal" - find path from initial position to "win position" - that find trajectory given boundary conditions
- Word2vec/BERT - "masked language modeling objective" - again predict trajectory given boundary conditions.



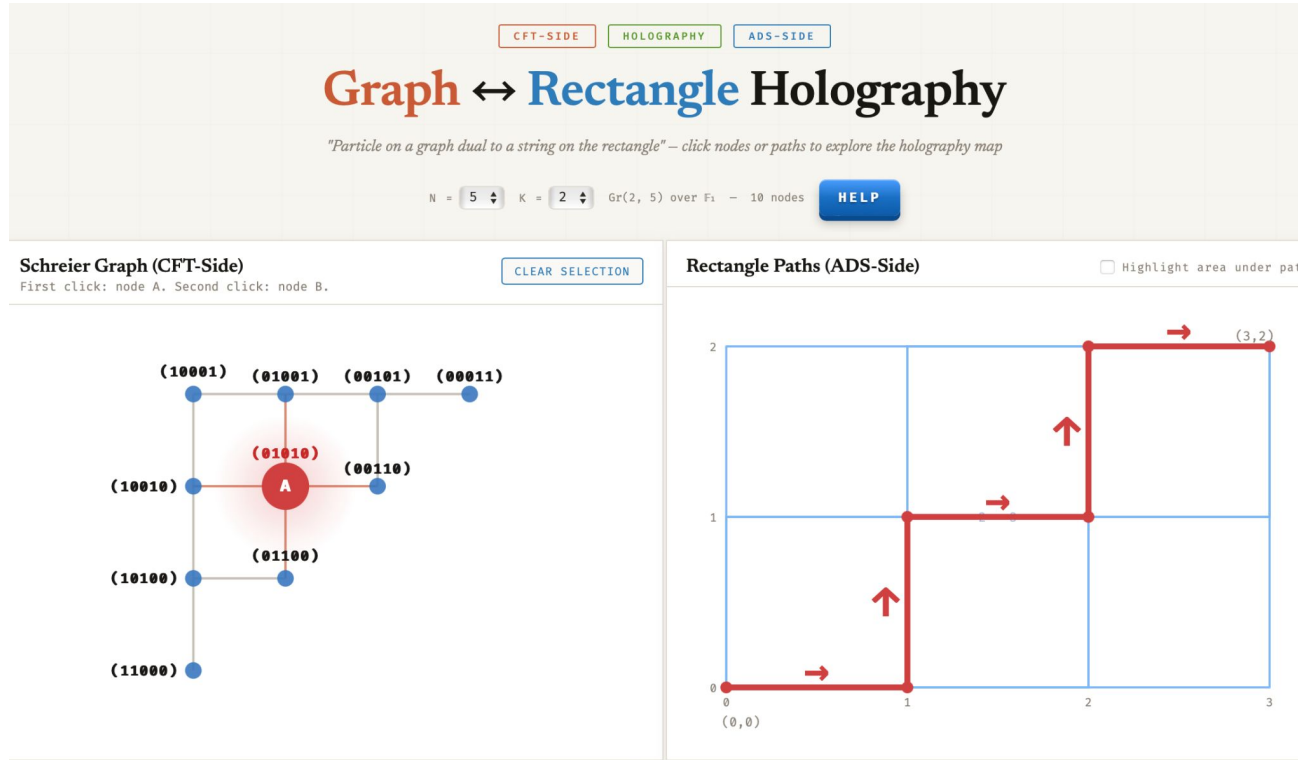
Describing trajectories on large graphs - not easy

Paths on such graphs - sentences in alphabet M_i :

E.g. Red.Blue.Red.Blue...

Or L.R.X.R.X.L.R.L.X...

Key idea - there exists a holographically dual description of particles in terms of strings - and that gives more tractable



There exists a dual object (e.g. rectangle) such that for any node of graph - corresponds a "line" (string) and description in terms of strings converts complicated tasks to more tractable

Complexity = Area (same as in AdS/CFT)

Graph ↔ Rectangle Holography

"Particle on a graph dual to a string on the rectangle" – click nodes or paths to explore the holography map

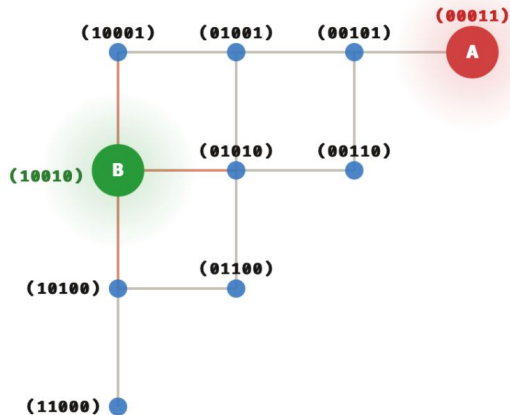
N = 5 K = 2 Gr(2, 5) over F₃ – 10 nodes

HELP

Schreier Graph (CFT-Side)

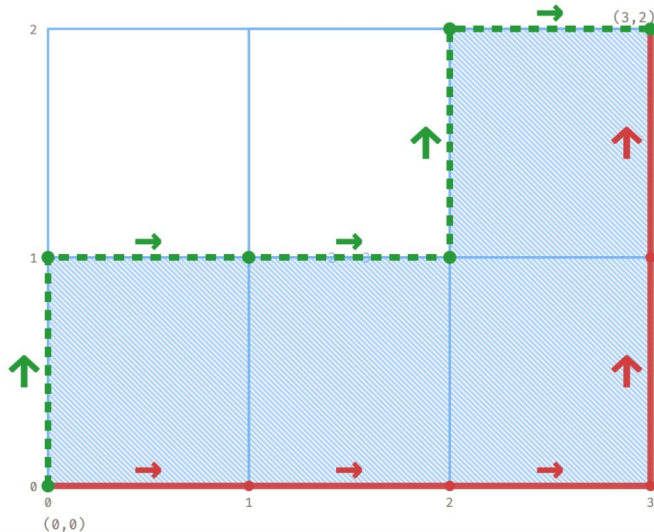
First click: node A. Second click: node B.

CLEAR SELECTION



Rectangle Paths (ADS-Side)

☐ Highlight area under path



For example -
difficult task -
compute length
of the shortest
path
("complexity") -
converts to
compute area
under the curve
- which is much
more tractable

Results and SOTA

1. C. Sims 1970: Schreier–Sims algorithm (permutation groups)
2. Knuth, Donald E. (1991): randomized version (10^{20} - 10^{30})
3. **CayleyPy beats by orders and orders of magnitude (10^{50} - 10^{120})**

Complexity:

1. 1981 NP-hard: optimal solutions and diameters
2. 1998-....: optimal solution
NP-complete for Rubik's cube, etc

n	GAP Length	Ideal Length $n(n-1)/2$	AI Length
9	41	36	36
10	51	45	45
11	65	55	55
12	78	66	66
13	99	78	78
14	111	91	91
15	268	105	105
16	2454	120	120
17	380	136	136
18	20441	153	153
19	3187	171	171
20	217944	190	190
21	-	210	210

Testbed for: automatic provers, "chatGPT", robotics ...

L.Babai conjecture - way to Fields medal :)

1. For any generators of S_n , diameter $< O(n^2)$ (open: even polynomial bound)
2. CayleyPy 10^2 - 10^3 times FASTER than GAP, so extensive comp.exp.:
3. **New**: explicit generators which conjecturally maximize diameter - follow simple pattern: "square with whiskers" + involutions
4. **New** conjecture: diameter $< n^2/2 + 4n$

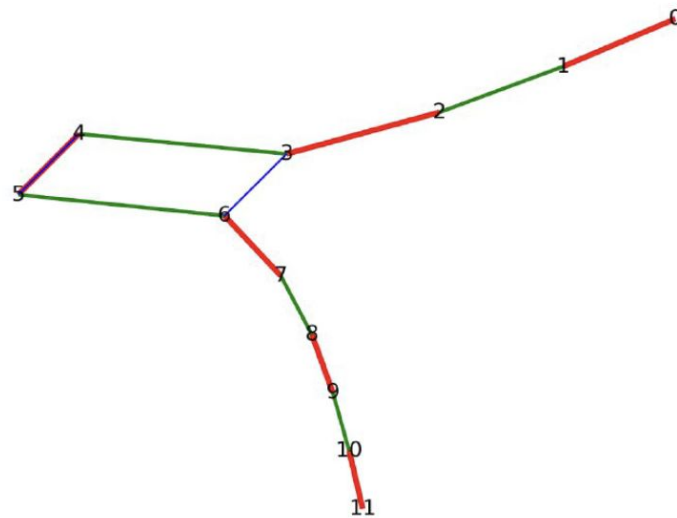
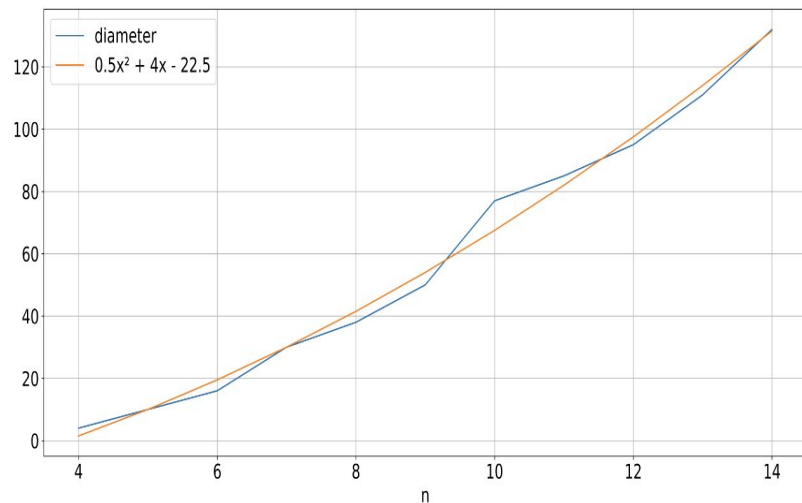


FIGURE 10. Maximal diameters found up to $n \leq 14$ and quadratic fit: $n^2/2 + 4n - 22.5$.

Surprise: Diameters - "findable"(?!) = quasi-polynomials

Example of quasi polynomial: $n^2/3$ if $n \equiv 0 \pmod 3$, $(n^2-1)/3$ if $n \equiv 1, -1 \pmod 3$

1. Conjecture 1: Diameters of S_n are quasi-polynomials (for constructive generators)
2. Conjecture 2: Length (word metric) of ANY element is quasi-polynomial (constructive element and constructive generators)

Example (<https://oeis.org/A039745>):

Two generators L - left cyclic shift, $X = (1,2)$

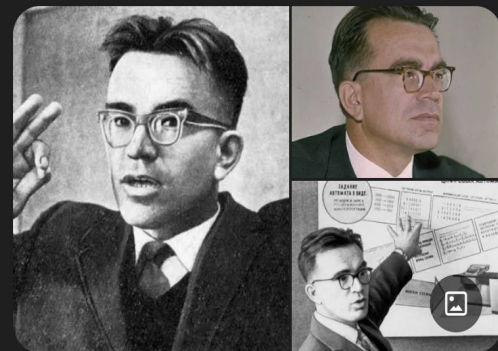
"father of Soviet cybernetics" V.M.Glushkov 1968 problem:

New conjecture: $n > 4$: diameter $(3n^2 - 8n + 9)/4$, n odd

$(3n^2 - 8n + 12)/4$, n even . Checked $n < 16$

Victor Glushkov

Soviet computer scientist

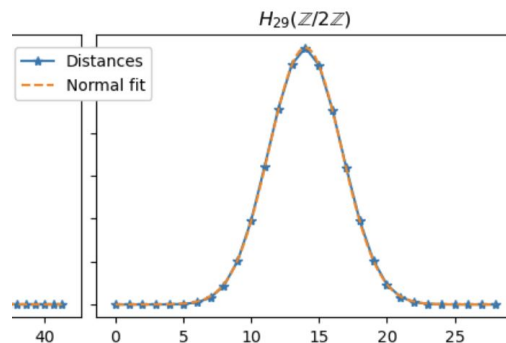
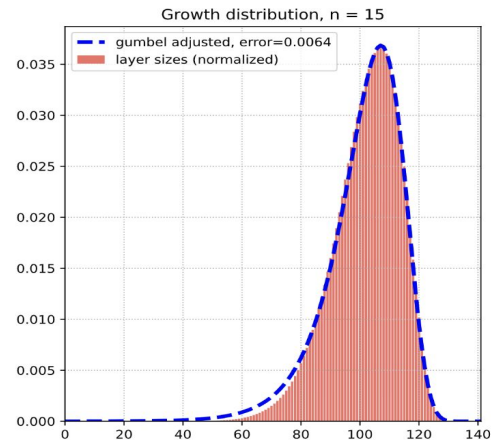


Victor Mikhailovich Glushkov was a Soviet computer scientist. He is considered to be the founding father of information technology in the Soviet Union and one of the founding fathers Soviet cybernetics.

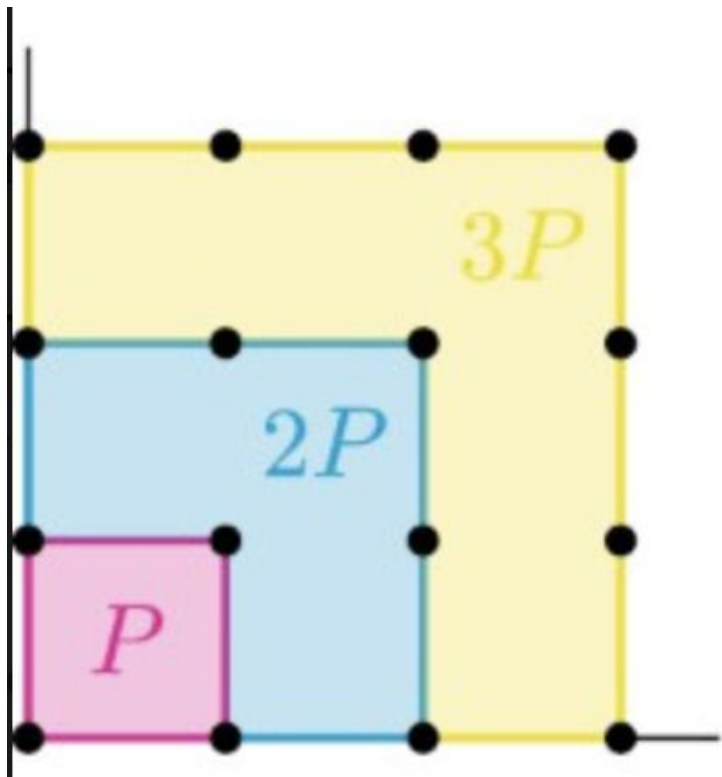
Hundreds conjectures ("life exists beyond diameters")

Black diamond - new conjecture, blue diamond - we can prove

Gene-	GroDia-	Growth								Anti-	Algo-	Met-Spec-	Mixing
rators	up meter	PDF	F-la	Mean	Mode	Var	Skew	Kurt	podes	rithm	ric	trum	Time
Coxeter	S_n	$\frac{n(n-1)}{2}$	Gauss	+	$\frac{n(n-1)}{4}$	$\frac{n(n-1)}{4}$	+	$\rightarrow 0$	$\rightarrow 0$	1	Bubble	+	?
Cyclic Coxeter	S_n	$\left\lfloor \frac{n^2}{4} \right\rfloor$	Gauss	?	$0.17(n^2 - n + 1)$	$\approx \text{Mean}$	?	$\rightarrow 0$	$\rightarrow 0$	$1 2$	?	+	Wig
LRX	S_n	$\frac{n(n-1)}{2} *$	Gumbel	?	$\approx 0.38n^2 - n$	$\approx 0.39n^2 - n$	?	$\rightarrow -0.7$	$\rightarrow 3.3$	?	?	?	Uni
LX-Glushkov	S_n	$\frac{3n^2 - 8n + 9}{4}$	Gumbel	?	$\approx 0.57n^2 - 2n$	$\approx 0.57n^2 - 1.6n$	?	$\rightarrow -0.7$	$\rightarrow 0.5$	*	?	?	?
LARX	S_n	$\frac{n^2 - 2}{2}$	?	?	$0.4n^2 - 0.7n$	$\approx \text{Mean}$	?	?	?	$+$	?	?	?
LARX+I	S_n	$\frac{n(n+6) - 12}{4}$	?	?	$\approx \frac{n(n+1)}{4}$	$\approx \text{Mean}$	?	?	?	$+$	?	?	?
LSL	S_n	$\frac{n(n-3)}{2} + 3$	?	?	$\approx 0.4n^2 - 1.5n$	$\approx \text{Mean}$	?	?	?	$+$	?	?	?
LSL+I	S_n	$\frac{n(n+4)}{4} - 3$	?	?	$\approx 0.2n^2$	$\approx 0.2n^2$	?	?	?	?	?	?	?
3-cyc	A_n	$\left\lfloor \frac{n}{2} \right\rfloor$	?	+	$\approx D - 0.5$	$D - 0.5$	?	?	?	+	+	+	Int
(0ij)	A_n	$\left\lfloor \frac{3(n-1)}{4} \right\rfloor$	?	?	$\approx 0.55n$	$\approx \text{Mean}$	?	?	?	?	?	?	?
(01i)	A_n	$\frac{3n-5}{2} + i^n + (-i)^n$	?	?	$\approx n - 2$	$n - 1$	?	?	?	?	+	?	?
(01i)I	A_n	$\left\lfloor \frac{3n-6}{2} \right\rfloor$	?	?	$\approx n + 1.25 \ln n$	$\approx \text{Mean}$	?	?	?	?	?	?	Int



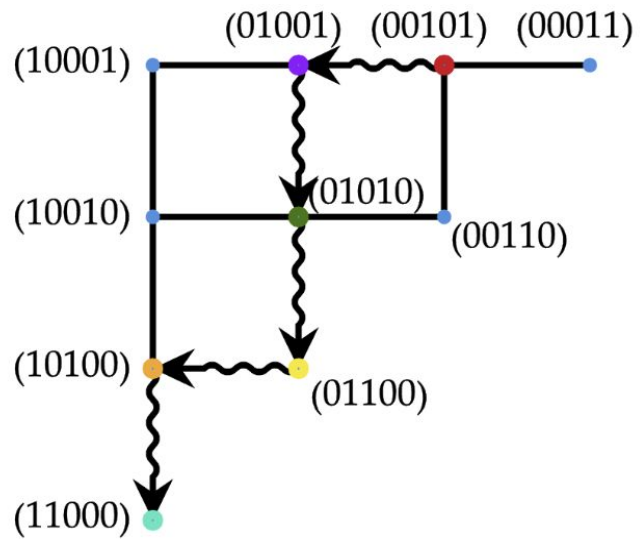
Ehrhart polynomial - main source of quasi-polynomials



Take a rational polygon (or polytope) count integer points in it and in its n -scaled copy - theorem of Ehrhart says values will be quasi-polynomials.

The only way to explain conjecture that diameters and word-metrics are quasi-polynomials - is to propose that there exist planar polygon and holography such that diameters/words metrics = numbers of integer points in/under the polygon/string

6	4	3
5	2	1



6	4	3
5	2	1

6	4	3
5	2	1

6	4	3
5	2	1

6	4	3
5	2	1

6	4	3
5	2	1

6	4	3
5	2	1

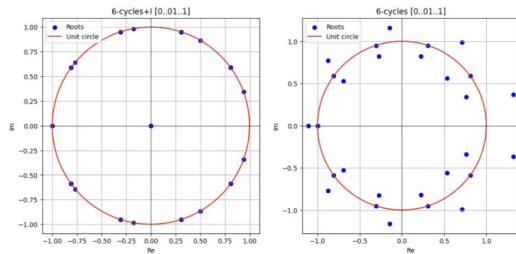


FIGURE 5. Roots of the H -polynomials in the complex plane. In the left panel, all non-trivial roots have modulus equal to one, so the “Riemann conjecture” holds, whereas in the right panel it is violated. Both panels correspond to k -consecutive cycle generators with $k = 6$: the left uses an inverse-closed generating set, while the right uses generators without inverses. These correspond to Schreier coset graphs for the subgroup $S_{\lfloor n/2 \rfloor} \times S_{n - \lfloor n/2 \rfloor}$. Whether the property that all roots have modulus one persists for larger k remains unclear.

Ehrhart series [\[edit\]](#)

We can define a [generating function](#) for the Ehrhart polynomial of an integral d -dimensional polytope P as

$$\text{Ehr}_P(z) = \sum_{t \geq 0} L(P, t) z^t.$$

This series can be expressed as a [rational function](#).

Specifically, Ehrhart proved (1962) that there exist complex numbers h_j^* , $0 \leq j \leq d$, such that the Ehrhart series of P is^[1]

$$\text{Ehr}_P(z) = \frac{\sum_{j=0}^d h_j^*(P) z^j}{(1 - z)^{d+1}}.$$

[Richard P. Stanley](#)’s non-negativity theorem states that under the given hypotheses, each h_j^* will be a non-negative integer, for $0 \leq j \leq d$.

4.3. **$SL(2, \mathbb{Z})$ and the Farey graph.** Here we reinterpret the results of [Camilo2019] in a way closer to the present exposition and to the principles of holography and “complexity = area”. It was argued there that to each element of $SL(2, \mathbb{Z})$ one may associate a curve in the hyperbolic plane whose enclosed area approximately corresponds to the complexity of the element with respect to the standard generators S and T . In this language, the map from a group element (i.e. a node of the Cayley graph) to such a curve may be viewed as a holography map, in agreement with our framework. The cited results therefore imply that a variant of the “complexity = area” principle holds, at least approximately. An illustration is shown in Figure 6.

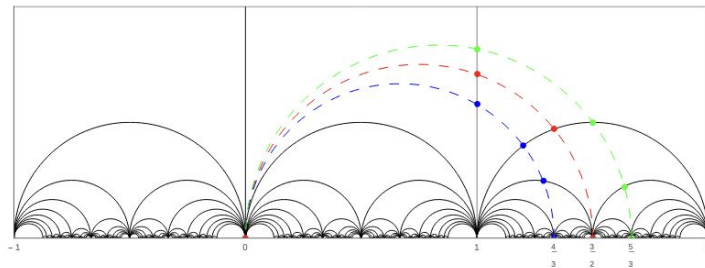


FIGURE 6. Farey graph on the upper halfplane. From [Camilo2019]. The dashed curves are “holography images” of some elements of $SL(2, \mathbb{Z})$ areas under them correspond to complexities of these elements. “Complexity = area” holds approximately.

CayleyPy1 approach - diffusion distance based:

- 1) **Train set via random walks:** Ran many random walks trajectories starting from identity “id” and store pairs: (g,k): g - node of graph, k - in how many steps it was achieved by random walk. (

Train set - pairs: (permutation , n_steps)

- 2) **Model:** Train machine learning model - to predict “k” from “permutation”

- 3) **Pathfinder (graph search, beam search):**

Start from node, take it all neighbours, compute neural net predictions, choose say 10 best (min predictions) nodes, repeat until "id" is found

Neural nets make mistakes: Beam Search to Compensate it

Beam Search = model of research community

Single researcher is exponentially less effective than two (group) of researchers, because "cannot break the wall" (stuck in local minimums)

Pipeline of all RL-like systems (e.g. AlfaGo):

**1 Neural Net says where to move
(trained by "playing with itself")**

2 Graph search compensates its mistakes (Monte-Carlo Tree Search, Beam Search, A-star, many other global optimization algorithms)

Beam search:

**Has a single parameter - size of the beam
(size of research group) say 10:**

1 Look on all neighbours - choose 10 best

2 Iterate

**Surpsie: 10 cannot be compensated by 10
hours of single "researcher", because you
stuck in local minimums and only others can
researchers can provide breakthroughs to
come out**

Polymath, Crowd-sourcing - attract 1000+ AI experts to math problem - easy - Kaggle just needs to "press one button"

CayleyPy RapaportM2 - solve math mystery from 1959

Help mathematicians to find the God's number and algorithm - develop method to solve/sort permutations using E.Rapaport transformations



#	Team	Members	Score	Entries	Last
1	Ruslan Grimov		120228357	3	8d
2	Alim Bijev		153013709	3	13d

CayleyPy Pancake sorting - outperform B.Gates

Prefix (pancake) sorting algorithm was proposed by B.Gates in 1979 - can you do better ?



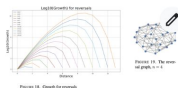
CayleyPy V.M.Glushkov permutations problem

Solve the problem by "father of Soviet cybernetics" V.M.Glushkov (1968)



CayleyPy Reversals - Estimate Evolution Distance

70 million years from human to mice - help to check that is true 🐹 Develop reversal sorting algorithms as optimal as possible



Kaggle = "world championship" for AI challenges. (It belongs to Google).

Math problems: optimize 1 number + easy to check correctness are perfect for Kaggle

Challenges attract THOUSANDS participants. Competition are extremely tough! Is not about money (no chance to get them, (I got twice :)). It is about:

- Adrenaline, fame/glory/honor
- "Reputation" - be "Kaggle grandmaster" - job will find you by itself

Create a challenge - if Kaggle decides "assign reputation" - you got 1000 AI-experts

Biological application - Cayley distance $\sim$ evolutionary distance

Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals

1999 cited: 1215

Sridhar Hannenhalli,  Pavel A. Pevzner

Authors:  [Sridhar Hannenhalli](#),  [Pavel A. Pevzner](#) [Authors Info & Claims](#)

Transforming men into mice (polynomial algorithm for genomic distance)

5.1.5. Transposons (block transpositions).

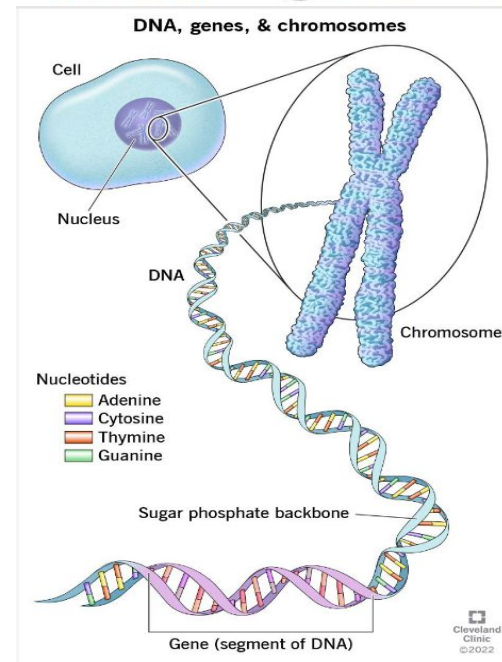
The generators called somewhat misleadingly in biological literature "transpositions" not to be confused with standard mathematical transpositions, better called "block transpositions", or we will call them just "transposons" - since their biological motivation is to describe "transposons" (moving fragments of genome). They can be described as follows: cutting out a segment and reinserting it elsewhere on the same chromosome (that segment is abstraction of biological transposon):

$$\dots u \underbrace{v \dots w}_{\Delta} x \dots y \dots \mapsto \dots u x \dots y \underbrace{v \dots w}_{\Delta} \dots$$

They are supported by CayleyPy under the name "transposons".

Conjecture.([OEIS-A065603](#)) Diameter of transposons is $\lceil \frac{n+1}{2} \rceil$ for $n \neq 13, 15$.

The conjecture formulated in [[ErikssonErikssonKarlanderSvenssonWästlund01](#)] (section 3 end), based on earlier results in [[BafnaPevzner98](#)].



Bill Gates theorem on sorting pancakes

- [Gates, W.; Papadimitriou, C. \(1979\). "Bounds for Sorting by Prefix Reversal". *Discrete Mathematics*. 27: 47–57. doi:10.1016/0012-365X\(79\)90068-2.](#)

In 1979, [Bill Gates](#) and [Christos Papadimitriou](#)^[11] gave an upper bound of $5/3n$

Diameter (=God's number) - the largest possible distance on the graph = the worst case number of operations for sorting algorithm = the number of moves solve in worst case

Pancake graph - Cayley graph for S_n , with generators: take first k pancakes, swap them and put back.

E.g. $k=4$: $(i_1, i_2, i_3, i_4, i_5, \dots) \rightarrow (i_4, i_3, i_2, i_1, i_5, i_6, \dots)$

Pancake Diameter is UNKNOWN - open problem ($n > 20$)

Diameter Rubik cube UNKNOWN $n > 3$. Etc...



CayleyPy Pancake sorting - outperform B.Gates

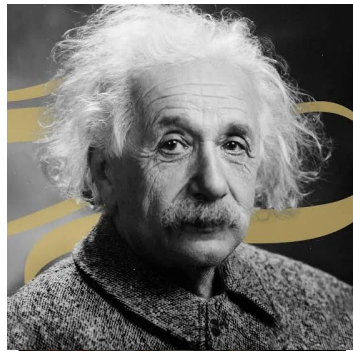
[Submit Prediction](#)[...](#)[Overview](#) [Data](#) [Code](#) [Models](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [Submissions](#) [Settings](#)

1	Tomas Rokicki		89573	2	3mo
2	KerbyA		89580	3	21m
3	Antonina Dolgorukova		89747	4	6h
4	Любовь Мунтян		91584	29	23d
5	Иван Литвак		107606	2	2d
6	Mikhail Seregin		122361	8	6d
7	Kirill Lazutkin		153201	4	3mo
8	Mehran Kazeminia		155733	43	4mo
9	Kutin Egor		157038	2	4mo

Thank you very much !

New ideas for AI models - string theory inspired

CayleyPy-4: AI-holography



The most powerful technique ever created by humankind for analysis of complex systems developed by generations of geniuses like Einstein and “Sheldon Cooper”

String duality: difficult problems -> more tractable

(“strong coupling” -> “weak coupling”)



Key example: Holographic AdS/CFT correspondence - “the most highly cited article in the field of high energy physics”

Quiz: what physics
Sheldon Cooper was
doing ? (“Big bang
theory”)

Finding: towards a way to apply it to AI tasks

What are the languages (e.g. protein) ? "Good embeddings" = strings

Proteins:

MQYLNKEDCNAMAFCAKMR...

Natural:

Hello world...

Reinforcement learning e.g. chess:

e2-e4; e7-e5;...

SMILES (molecules)

CC1CCC/C(C)=C1/C=C/C(C)=C/C...

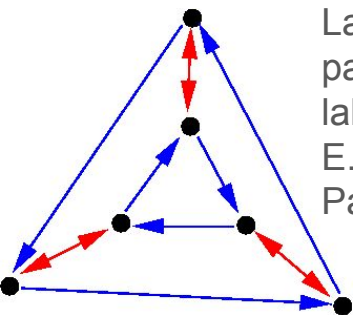
Any language can be defined by set of all texts in it - all sequences

Sequence embeddings - key to everything.

They transform initial representation to such that representation that downstream quantities can be computed easily, e.g.: "King-Man+Woman = Queen". So it is like string duality.

So "good embeddings" = strings via holography

Example. Holography: ROC-curves = strings, dual to nodes of certain graph. Mann-Whitney = Area under ROC curve => "strong->weak" coupling



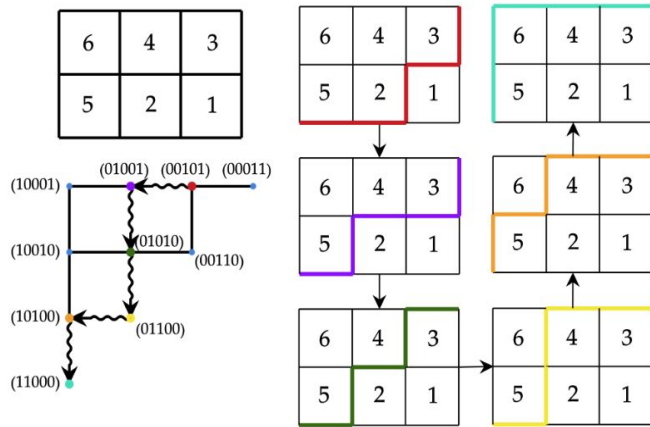
Languages as particle physics on graphs

Languages (texts) = particle trajectories on edge labeled graphs

E.g. - R(ed)/B(lue) edges.

Paths = words E.g. RBBRBR...

AI tasks = predict trajectory with initial or boundary conditions



Further string stuff:
Downstream computations require curved metrics, finite size corrections, etc