Machine learning and AI in Combinatorics Research

or (Machine learning for the working mathematician)

or (How I learned to stop worrying and love the machine)

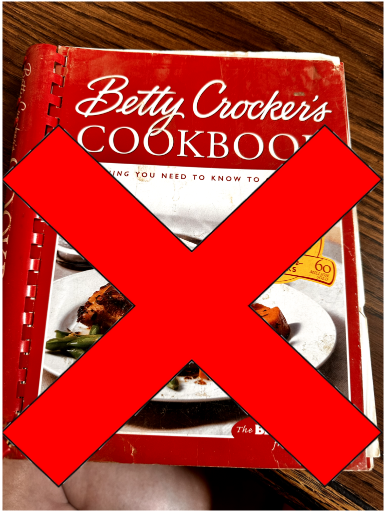
Kymani Armstrong-Williams, Edward Hirst, **Blake Jackson**, Kyu-Hwan Lee, Xiaoyu (Coco) Huang

ICERM 2025: Category Theory, Combinatorics, and Machine Learning

Outline

1. Learning combinatorial invariants (classification problems) + brute force
   1.1 Background on quivers and mutation
   1.2 Machine learning mutation-acyclicity of quivers
2. Learning combinatorial algorithms (mechanistic interpretability)
   2.1 q,t-Catalan numbers and a some Dyck path statistics
   2.2 Machine learning the Zeta map
3. A possible next step: Generating combinatorial bijections or algorithms

What I would like you to take away from this talk



Step 0: Generate data

To utilize big data techniques, you need big data*!

- **Cluster algebras**:
  - $Q = 1 \to 2 \quad \longrightarrow \quad \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \longrightarrow \quad [0, 1, -1, 0]$
  - $\dfrac{x_1x_2 + x_3x_4 + x_1x_5}{x_2x_4} \quad \longrightarrow \quad [[2, 1, 1, 1, 1], [0, 1, 0, 1]]$
- **Classical combinatorial objects**:
  - Dyck paths $\longrightarrow$ binary sequences, area sequences, etc.; possibly paired with statistics
  - SSTY $\longrightarrow$ (partition shape, reading word, q-statistic)
- **Category/representation theory**:
  - Modules over an algebraically closed field $\longrightarrow$ linear algebra
  - Length of a composition series, # simple modules arising as submodules, etc.

Brute force + Classification problems

Machine-learning mutation-acyclicity of quivers

Joint with Kymani Armstrong-Williams, Edward Hirst, and Kyu-Hwan Lee

Why do we want to study this?

Quivers define cluster algebras.

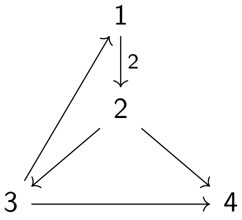
Mutation-acyclicity of a quiver has many desirable consequences for the cluster algebra:

- The resulting cluster algebra is finitely generated (José Simental's talk)
- The cluster algebra admits a green-to-red sequence
- There is a beautiful categorification of cluster algebras whose initial seed is acyclic (BMRRT 2006)

Quivers

Definition 1

A *quiver* $Q = (Q_0, Q_1)$ is a finite, directed multigraph without oriented 1- or 2-cycles. The number of vertices $|Q_0| = n$ is the *rank* of Q. We say a quiver is *acyclic* if it contains no oriented cycles of any length.

$Q =$ 

Mutation

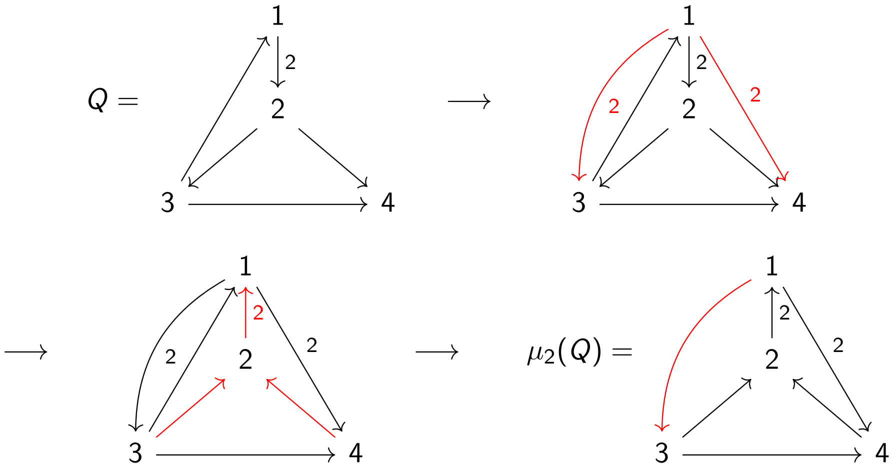
Definition 2

Given a quiver Q and a vertex k, we can define a new quiver $\mu_k(Q)$, called the *mutation of* Q *at vertex* k, in the following way:

1. for each directed 2-path $i \to k \to j$ in Q, add an arrow $i \to j$ in $\mu_k(Q)$,
2. reverse the direction of all arrows incidence to k,
3. pairwise delete any oriented 2-cycles in $\mu_k(Q)$ which have appeared as a result of step 1.

We say a quiver Q' is *mutation-equivalent* to Q if there is a sequence of vertices $[k_1, k_2 \ldots, k_\ell]$ such that $Q' = \mu_{k_\ell}(\cdots \mu_{k_2}(\mu_{k_1}(Q)) \cdots)$. We say Q is *mutation-acyclic* if it is mutation-equivalent to an acyclic quiver.

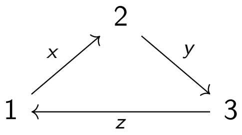
Mutation example



What we knew before: rank 1, 2, and 3 classifications

Remark 1 (Rank 1 and 2)

No loops or oriented 2-cycles $\implies$ trivially acyclic.

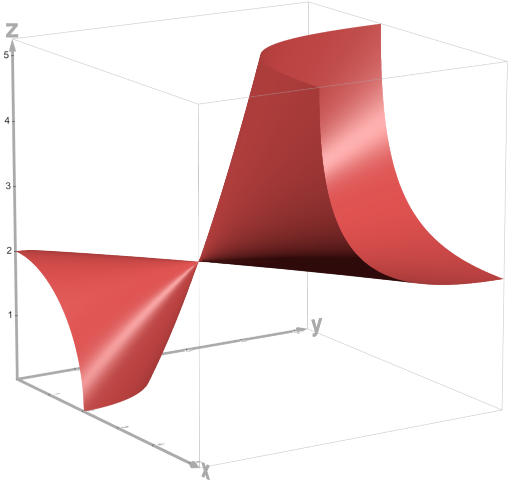


Theorem 1 (Beineke, Brüstle, and Hille 2011)

$C(x, y, z) = x^2 + y^2 + z^2 - xyz$ is a mutation invariant constant, called the Markov constant.

1. $C(x, y, z) > 4$ *or* $\min\{x, y, z\} < 2 \implies$ *Mutation-acyclic*
2. $C(x, y, z) \leq 4$ *and* $x, y, z \geq 2 \implies$ *Non-mutation-acyclic*

The geometry of the Markov constant: $C(x, y, z) = 4$

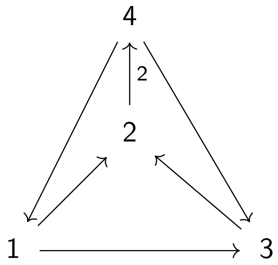


Known results in rank 4

Proposition 1 (Buan, Marsh, Reiten 2008)

Let Q be any quiver. If any full subquiver of Q is non-mutation-acyclic, then Q is non-mutation-acyclic.

There are also some interesting pathological quivers, like the "dreaded torus", in which all proper subquivers are mutation-acyclic but T is non-mutation-acyclic:



Open Problem 1

Prove Proposition 1 combinatorially.

Rank 4 progress using brute force

Theorem 2 (Armstong-Williams, Hirst, J, Lee 2025)

Suppose we have a quiver Q on 4 vertices with arrow weights taking on values of 0, 1, or 2. Then Q is non-mutation-acyclic if and only if one of the following conditions holds.

1. *Q contains a quiver in its mutation class with a rank 3 non-mutation-acyclic full subquiver.*
2. *Q is isomorphic to the dreaded torus quiver.*

Proof

The computer checked all 15,625 quivers. Surprisingly, some quivers took as many as 12 mutations to find an acyclic quiver in the mutation class!

Neural networks on a large dataset

Class	**Number of Matrices**	**Class Label**
MA	236142	0
NMA	184785	1

Run	**MCC**	**Test Accuracy Percentage**
1	0.731	86.3%
2	0.754	87.4%
3	0.745	86.9%
4	0.748	87.1%
5	0.759	87.6%
Average	0.747 ± 0.004	87.0 ± 0.2 %

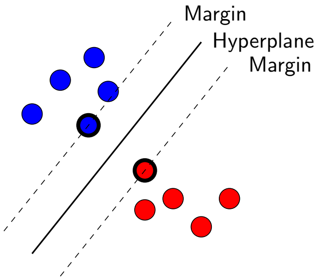
Remark 2

We did not do any mechanistic interpretation. Stay tuned! (:

Attempting to replicate the Markov constant

Geometric classification of Beineke, Brüstle, and Hille using separating hypersurfaces

$\implies$ try Support Vector Machines to do a geometric classification:

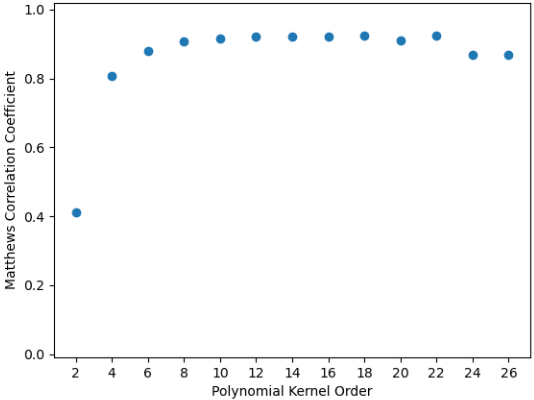


Remark 3

SVM = linear separating hypersurface.

Kernel trick $\longrightarrow$ nonlinear surface given by $h_{order}(features) \implies \binom{n+k-1}{k}$ "features."

Classifying rank 4 quivers: SVM performance



Remark 4

Data set = 15,625 quivers. The odd kernels have zero accuracy. The 8th order polynomial kernel contains 1,288 terms.

Finally, some interpretability

Here's the hypersurface equation for the kernel of order 2:

$$
\begin{aligned}
19.81x_1^2 - 14.41x_1x_2 + 19.81x_2^2 - 14.41x_1x_3 - 14.41x_2x_3 + 19.81x_3^2 + 14.41x_1x_4 \\
- 14.41x_2x_4 - 0.001x_3x_4 + 19.81x_4^2 + 14.41x_1x_5 - 0.001x_2x_5 - 14.41x_3x_5 \\
- 14.41x_4x_5 + 19.81x_5^2 - 0.002x_1x_6 + 14.41x_2x_6 - 14.41x_3x_6 + 14.41x_4x_6 \\
- 14.41x_5x_6 + 19.81x_6^2 = 276.49
\end{aligned}
$$

Remark 5

It's not entirely unlike the Markov constant, but still quite far off.

Future directions

- Do some mechanistic interpretation!
- Leverage the separating hypersurfaces to look for “Markov like” invariants.
- Wait for a human (Scott Neville) to discover something new that we can use.
- Transformers?
- Throw it into Funsearch/AlphaEvolve and see what happens (or doesn’t happen)?

Learning combinatorial algorithms (bijections)

Machine-learning the zeta map (work in progress)

Joint with Coco Huang and Kyu-Hwan Lee

q,t-Catalan numbers: combinatorial perspective

The q,t-Catalan numbers originally arose as the bi-graded Hilbert series of the diagonal coinvariant algebra

$$DR_n = \mathbb{C}[X_n, Y_n]/\langle \sum_{i=1}^{n} x_i^h y_i^k, \forall h + k > 0 \rangle$$

$$C_n(q,t) = \mathcal{H}(\ (DR_n^*)^{\epsilon}; q, t).$$

Theorem 3

There exists nonnegative statistics area, bounce, and dinv on Dyck paths such that

$$C_n(q,t) = \sum_{\pi \in \mathcal{D}(n)} q^{area(\pi)} t^{bounce(\pi)} \qquad \text{(Haglund 2003)}$$

$$= \sum_{\pi \in \mathcal{D}(n)} q^{dinv(\pi)} t^{area(\pi)} \qquad \text{(Haiman 2000)}$$

The zeta map: a rich history

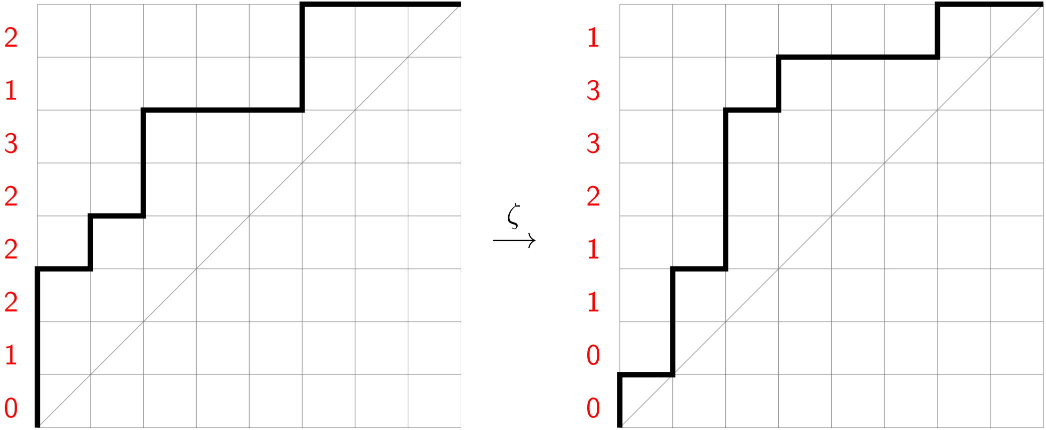
The zeta map is a bijection on Dyck paths of length $2n$ satisfying the property

$$(\mathit{area}(\pi), \mathit{bounce}(\pi)) = (\mathit{dinv}(\zeta(\pi)), \mathit{area}(\zeta(\pi))).$$

Since its definition, there have been many combinatorial descriptions of the zeta map and extensions to other settings:

- G. Andrews, C. Krattenthaler, L. Orsina, P. Papi (2002)
- J. Haglund (2003)
- N. Loehr (2003)
- E. Egge, J. Haglund, K. Killpatrick, D. Kremer (2003)
- E. Gorsky and M. Mazin (2014)
- K. Lee, L. Li, N. Loehr (2014)
- D. Armstrong, N. Loehr, G. Warrington (2015)
- H. Thomas and N. Williams (2018)

The zeta map in action



IDEA:

Since it is so well studied and understood with multiple descriptions, apply ML techniques to learn the zeta map and try to interpret the trained models. In other words, this is the perfect playground for mechanistic interpretability.

Our dataset: $[\ \pi\ ,\ \zeta(\pi)\ ]_{\pi \in \mathcal{D}(13)}$ where π is the binary sequence repn of a Dyck path.

Transformers

Jay Alammar's "The Illustrated Transformer"

What we think the transformer learned

Based on what we know about the descriptions of the zeta map given by humans, we believe that the transformer has learned a variant of the "sweep map."

Definition 3

For a Dyck path $\pi \in \{N, E\}^{2n}$, we can define the level vector recursively as follows:

$$\ell_0 = 0 \qquad \ell_i = \begin{cases} \ell_{i-1} + 1 & \text{if } \ell_i = N \\ \ell_{i-1} - 1 & \text{if } \ell_i = E. \end{cases}$$

The sweep $sw^{-}_{1,-1}(\pi)$ of π is defined as follows:

- Start at level -1 and sweep across the word of π from right to left, recording each step that has level -1 into $sw^{-}_{1,-1}(\pi)$ as you go.
- Repeat for $\ell = -2, -3, \ldots, 4, 3, 2, 1, 0$. The resulting word is $sw^{-}_{1,-1}(\pi)$.

An example:

Consider the following Dyck path with levels

$$\pi = N\ N\ E\ N\ E\ E\ N\ E$$
$$\ell = 1\ \ 2\ \ 1\ \ 2\ \ 1\ 0\ \ 1\ \ 0$$

Then the sweep of π will be

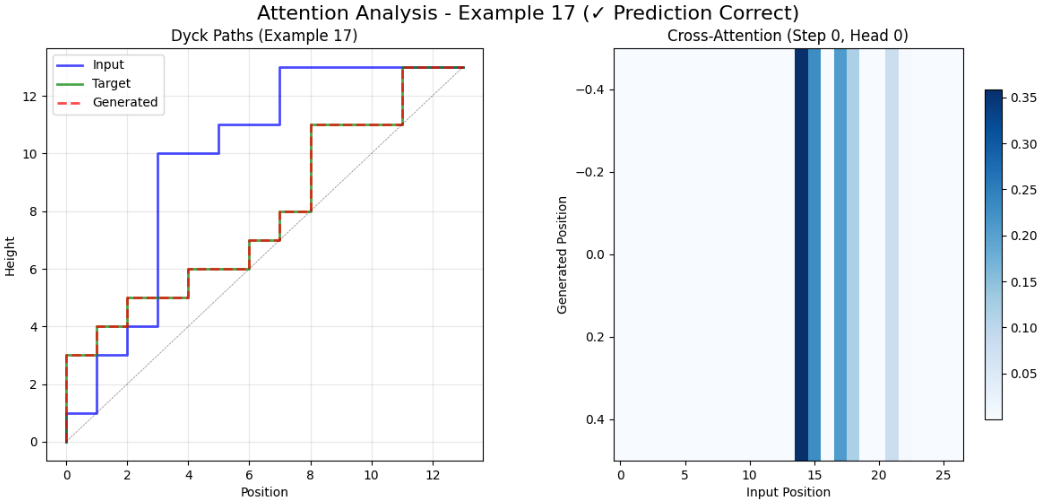
$$sw^{-}_{1,-1}(\pi) = N\ N\ N\ E\ E\ N\ E\ E$$

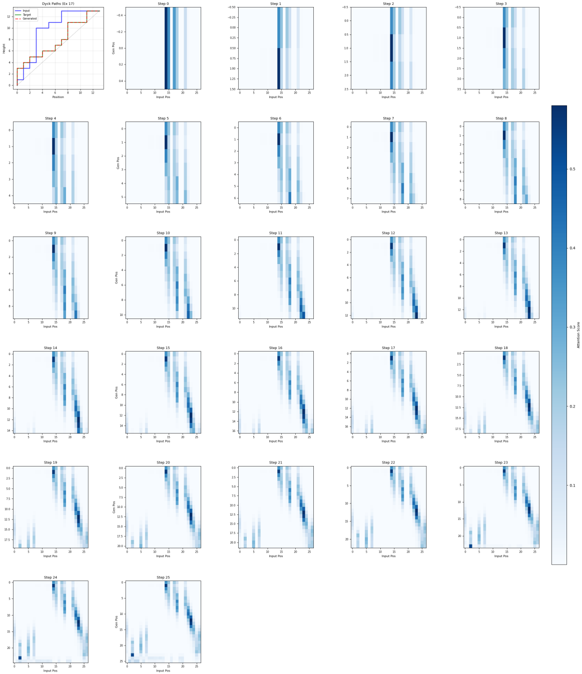
Supporting evidence

We have lots of reasons to think this is the case:

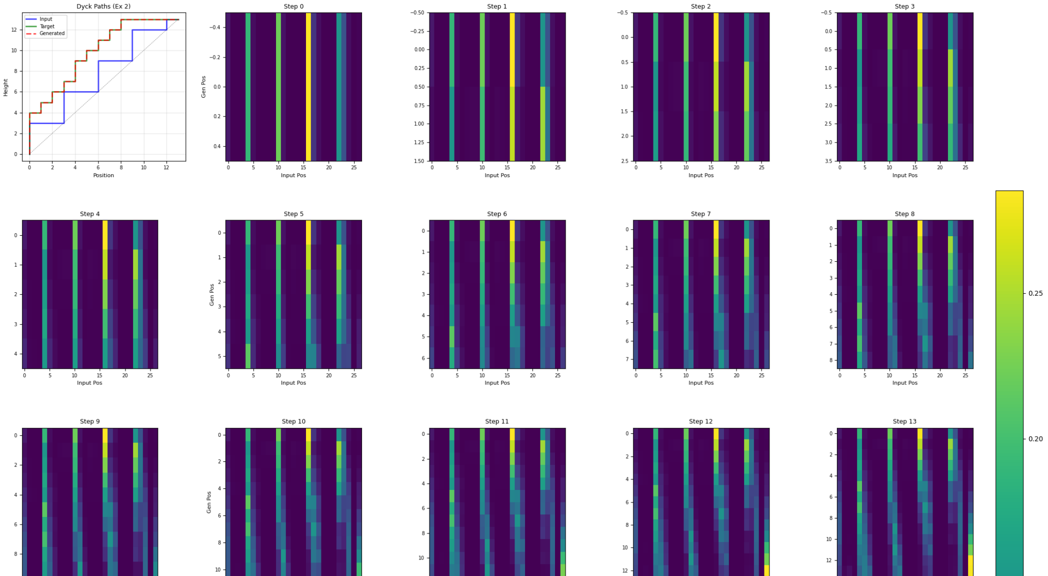
- Encoder embeddings strongly encode level information (98% accuracy via logistic regression)
- The observed attention patterns across multiple examples match a version this algorithm (we have yet to implement systematic probing) (pictures on the next slides)
- (!) The "N" moves appear unnecessary* in the cross-attention phase: manually killing attention to them still yields nearly 100% accuracy on predicting the first ~ 15 steps of a 26 step Dyck path.
    - *The encoder states for the "E" steps do retain information about any preceding steps (as confirmed by probing).

Supporting evidence

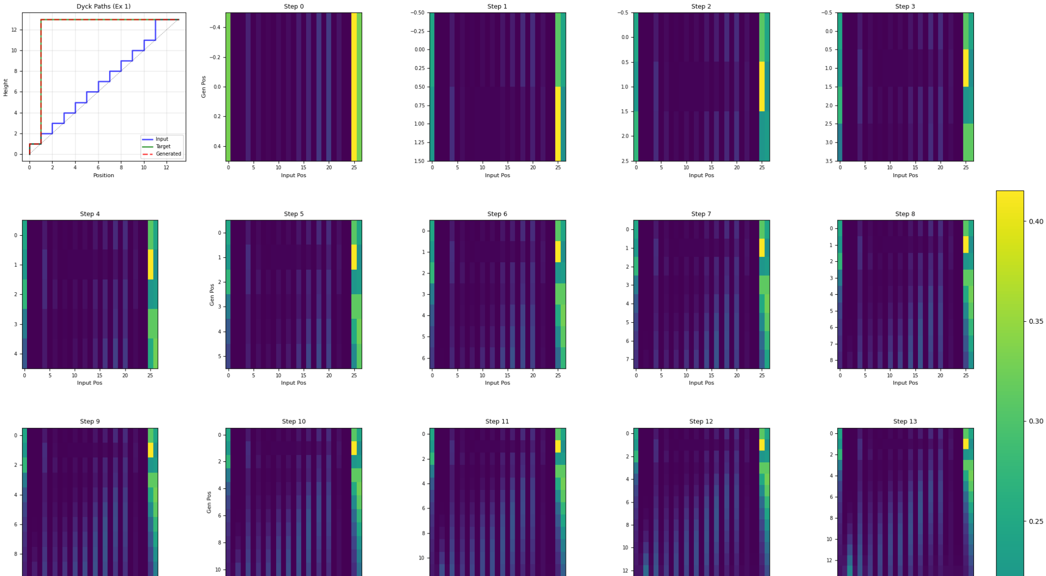



Dyck Paths (Ex 17)
Input
Target
Generated
Height
Position
Step 0
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Step 7
Step 8
Step 9
Step 10
Step 11
Step 12
Step 13
Step 14
Step 15
Step 16
Step 17
Step 18
Step 19
Step 20
Step 21
Step 22
Step 23
Step 24
Step 25
Gen Pos
Input Pos
Attention Score

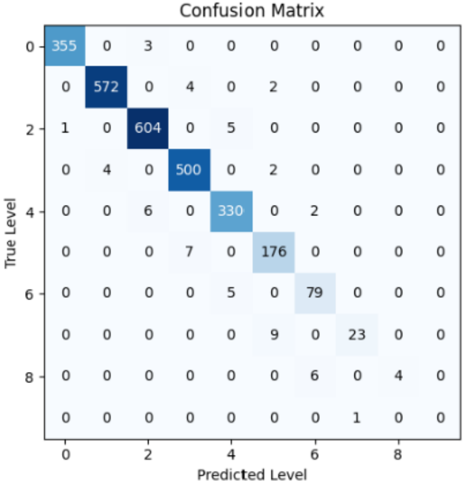
Cross-Attention Head 0 for Example 2 (✓ Correct)

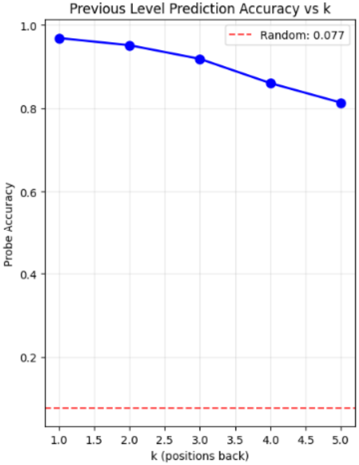


Cross-Attention Head 0 for Example 1 (✓ Correct)



The confusion matrix for the levels of a Dyck path





Probing indicates that encoder states remember their levels and remember previous positions' levels, but fuzzily.

Not pictured

We also plan to attempt to evolve a zeta map using genetic AI tools like FunSearch or AlphaEvolve. We will try different initial programs paired with different hyperparameters to test what works and what doesn't.

The future: generating combinatorial algorithms?

Joint with ...? (This could be you)

The hard part

Until better methods are developed, we really only have two methods for automatically creating combinatorial bijections from nothing:

- "Piece of straw in the haystack" followed by mechanistically interpretable ML
  - A bijection satisfying constraints can almost always be obtained by some (stochastic) greedy algorithm
- Genetic AI: FunSearch, AlphaEvolve, etc.
  - Jordan spoke about this earlier in the week, but success stories using these methods to make progress on "difficult" problems are few and far between (hopefully this will change when Terry Tao finishes his writeup)
  - I remain optimistic