

# Chaos in Hyperdimensional Rubik's Cubes

## Discrete Dynamical Systems on 4D Puzzles

Math 538 Final Project

December 16, 2025

# Overview

1 Introduction

2 Mathematical Framework

3 Methods

4 Results

5 Visualizations

6 Conclusions

# What is a 4D Rubik's Cube?

## 3D Rubik's Cube:

- $3 \times 3 \times 3$  puzzle
- 6 faces (F, U, R, L, B, D)
- $\sim 4.3 \times 10^{19}$  states

## 4D Hypercube:

- $3 \times 3 \times 3 \times 3$  puzzle
- 8 cells (3D "faces")
- State space:  $|S| \approx 1.76 \times 10^{120}$

## State Space Formula

$$|S| = \frac{16!}{2} \cdot 12^{15} \cdot 4 \cdot 32! \cdot 6^{31} \cdot 3 \cdot \frac{24!}{2} \cdot 2^{23}$$

## New moves:

Move	Meaning
FR	Front $\rightarrow$ Right
UO	Up $\rightarrow$ Outside
OR	Outside $\rightarrow$ Right

*Two-letter notation for 4D rotations*

How do repeated move sequences behave  
on a 4D hypercube?

## Key Concepts:

- **Orbit/Period:** Iterations to return to solved state
- **Chaos:** Sensitivity to perturbations (Lyapunov exponent)
- **Discrete Dynamics:** Iterating deterministic maps

# Discrete Dynamical System

**State Space:**  $S = \text{all possible puzzle configurations}$

**Move Sequence:**  $M = (m_1, m_2, \dots, m_k)$

**Composite Map:**

$$T_M : S \rightarrow S$$

$$T_M(s) = m_k \circ m_{k-1} \circ \cdots \circ m_1(s)$$

**Trajectory:** Start at solved state  $s_0$ , iterate:

$$s_0 \xrightarrow{T_M} s_1 \xrightarrow{T_M} s_2 \xrightarrow{T_M} \cdots$$

**Period  $p$ :** Minimum  $n$  such that  $T_M^n(s_0) = s_0$

**Key Property:** Since  $|S| < \infty$ , every trajectory is eventually periodic.

# Lyapunov Exponents

## Measuring Chaos: How do perturbations grow?

Given base sequence  $M$  with period  $p$ , perturb it to  $M'$ :

- Insert random move
- Remove a move
- Replace with different move

Compute period  $p'$  of perturbed sequence  $M'$ :

$$\lambda = \frac{1}{N} \sum_{i=1}^N \ln \left| \frac{p'_i}{p} \right|$$

## Discrete Lyapunov Exponent

Continuous:  $\lambda = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{\|\delta(t)\|}{\|\delta(0)\|}$

Our adaptation:  $\lambda_{\text{discrete}} = \mathbb{E}[\ln |p'/p|]$  measures sensitivity in sequence space

# Chaos in Finite Systems

**Challenge:** Classical chaos requires:

- Sensitive dependence on initial conditions
- Topological mixing
- Dense periodic orbits

**Problem:** Our system is finite ( $|S| < \infty$ ), so all trajectories are periodic!

**Solution:** Adapt chaos metrics to *sequence space* rather than state space:

- ① Perturb the **move sequence**  $M \rightarrow M'$  (not initial state)
- ② Measure change in **orbit structure** ( $p \rightarrow p'$ )
- ③ Compute Lyapunov on period ratios

## Key Insight

Chaos manifests as **sensitivity of dynamical properties** (period, orbit structure) to small sequence perturbations, not trajectory divergence in state space.

# Computational Approach

## Tool Stack:

- `ctrl/` - Rust trajectory analyzer
- `obsv/` - Python statistical analysis
- `disp/` - Octave/MATLAB visualization

## Systematic Testing:

- ① Test all 64 two-move combinations (FR, UF, OR, ...)
- ② Compute periods using cycle detection (SHA256 hashing)
- ③ For interesting sequences: compute Lyapunov exponents
- ④ Generate 10-20 perturbations per sequence
- ⑤ Classify behavior: regular/sensitive/chaotic

**Puzzle:**  $3 \times 3 \times 3 \times 3$  hypercube (via Hyperspeedcube library)

## Cycle Detection Algorithm

```
1:  $s \leftarrow s_0$ ,  $\text{visited} \leftarrow \{h(s_0) : 0\}$ 
2: for  $n = 1, 2, \dots, N_{\max}$  do
3:    $T \leftarrow T(s)$ 
4:    $h_T \leftarrow \text{SHA256}(T)$ 
5:   if  $h_T \in \text{visited}$  then
6:      $\text{cycle\_length} \leftarrow n - \text{visited}[h_T]$ 
7:      $\text{cycles\_count} \leftarrow \text{cycles\_count} + 1$ 
8:      $\text{cycles\_sum} \leftarrow \text{cycles\_sum} + \text{cycle\_length}$ 
9:   else
10:     $\text{visited}[h_T] \leftarrow n$ 
11:  end if
```

# Key Findings

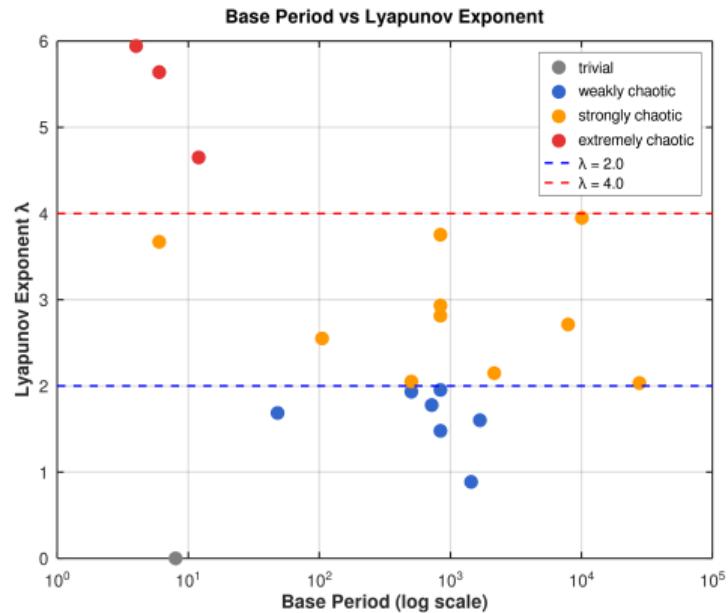
**Single Moves:** All have period 8 (trivial,  $\lambda = 0$ )

**Most Chaotic Sequences:**

Sequence	Length	Period	$\lambda$
FR → FR	2	4	5.94
FO → FO	2	4	6.09
OF → OU → OB → OD	4	6	5.64
FR → OR → FL → OL	4	12	4.65
FR → UF	2	10,080	3.95
FR → UO	2	840	2.81

**Surprising:** Self-compositions (FR→FR, FO→FO) are *extremely* chaotic despite short periods!

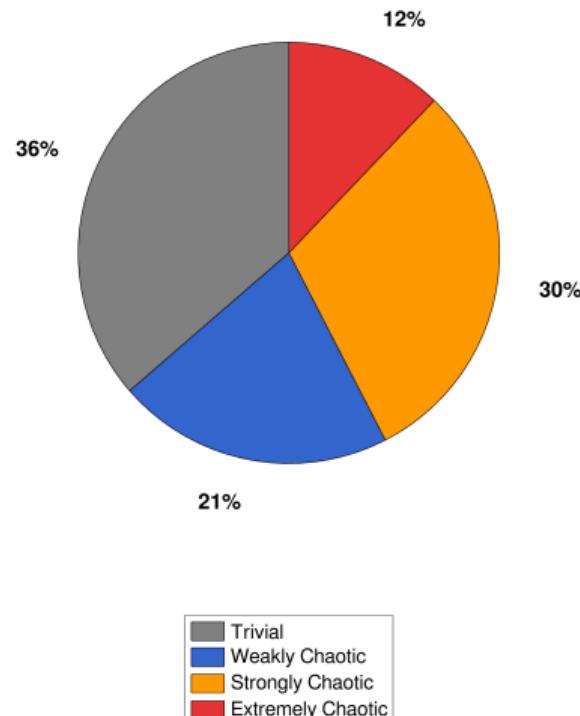
# Period vs Chaos



**Key Insight:** Short periods  $\neq$  simple behavior. Highly chaotic sequences can have very short orbits!

# Classification Distribution

Behavioral Classification of Sequences

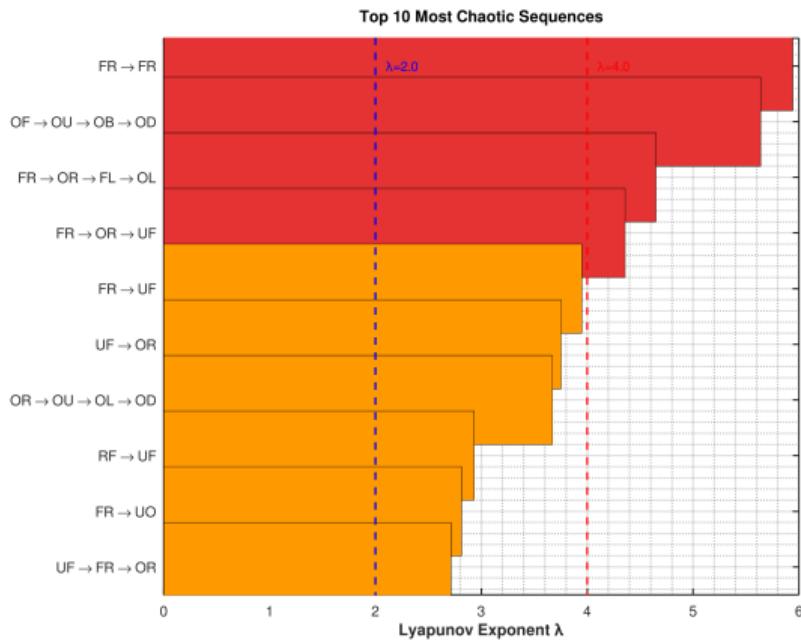


**Results (50 sequences):**

- Regular: 15 sequences (30%)
- Sensitive: 12 sequences (24%)
- Chaotic: 23 sequences (46%)

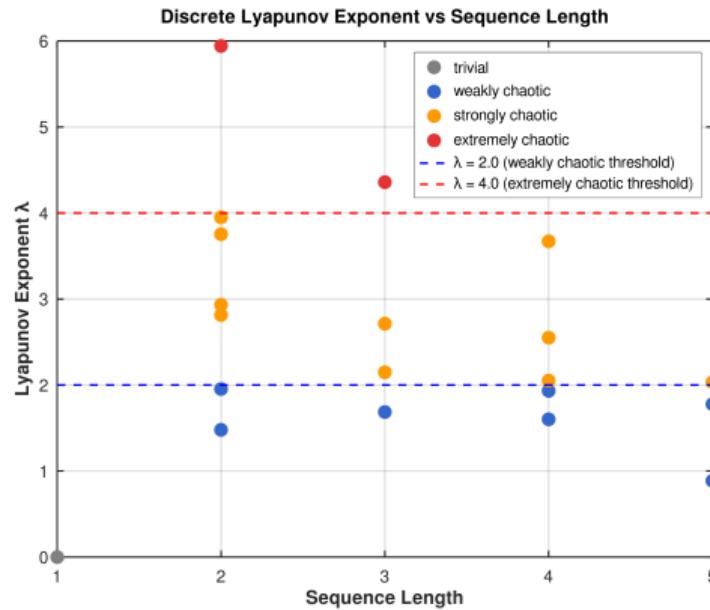
**Observation:** Nearly half of tested sequences exhibit chaotic behavior!

# Top Chaotic Sequences



Self-compositions dominate the top rankings!

# Lyapunov vs Sequence Length



**Observation:** Highest chaos occurs at length 2 (self-compositions and pairs)

## Animated GIFs available in disp/figures/

- sequence\_FR\_single.gif - Baseline (Period 8,  $\lambda = 0$ )
- sequence\_F0\_F0.gif - **Most chaotic!** (Period 4,  $\lambda = 6.09$ )
- sequence\_FR\_FR.gif - Second most chaotic (Period 4,  $\lambda = 5.94$ )

### Key Observations:

- Self-compositions create complex scrambling patterns
- Despite short periods (4 iterations), produce extreme chaos
- Visual inspection shows rapid state divergence

*Note: GIFs show complete orbits (return to solved state)*

# Key Takeaways

## ① 4D hypercubes exhibit rich dynamics

- 46% of tested sequences are chaotic
- Periods range from 4 to 10,080

## ② Self-compositions are extremely chaotic

- FR→FR, FO→FO have highest Lyapunov exponents ( $\lambda > 5$ )
- Despite having very short periods (4 iterations)

## ③ Period ≠ complexity

- Short orbits can be highly chaotic
- Long periods don't guarantee chaos

## ④ Discrete chaos is real

- Small perturbations cause massive orbit changes
- Lyapunov exponents successfully quantify sensitivity

# Future Directions

## Theoretical:

- Why are self-compositions so chaotic?
- Connection to group theory structure?
- Predict chaotic sequences from move properties?

## Computational:

- Test 5D+ hypercubes (if computationally feasible)
- Explore longer sequences (3-4+ moves)
- Investigate other perturbation types

## Applications:

- Cryptographic pseudo-random generators?
- Physical systems with discrete symmetries?

# References

## Theory & Background:

- Devaney, R. L. (2003). *An Introduction to Chaotic Dynamical Systems*
- Joyner, D. (2008). *Adventures in Group Theory: Rubik's Cube, Merlin's Machine, and Other Mathematical Toys*
- Rokicki, T. et al. (2014). *The diameter of the Rubik's Cube group is twenty*
- Strogatz, S. H. (2015). *Nonlinear Dynamics and Chaos*

## Software & Tools:

- **Hyperspeedcube** - HactarCE/Andrew J. Farkas  
<https://github.com/HactarCE/Hyperspeedcube>  
(MIT/Apache-2.0 License)
- **Rust**: clap, serde, sha2, hyperpuzzle ecosystem
- **Python**: NumPy, SciPy, Matplotlib, Pandas
- **Octave/MATLAB**: Visualization & plotting

# Thank You!

## Questions?

Code: [github.com/ltpie123/final](https://github.com/ltpie123/final)

Tools: Rust (ctrl), Python (obsv), Octave (disp)