Statistical Analysis of Texas 42 Game Trees

For Statistical Review

This report presents a structural analysis of exhaustively-solved Texas 42 domino game trees. We have computed exact minimax values for millions of game states across hundreds of random deals, producing a complete dataset of perfect-play outcomes.

We seek statistical guidance on: 1. Better methods for characterizing the value function's structure 2. Approaches we may have missed for dimensionality reduction 3. Statistical tests for the significance of our findings 4. Alternative framings that might reveal hidden structure

---

The Data

Texas 42 is a four-player partnership trick-taking game using a double-six domino set (28 dominoes). One team "declares" (bids and names trump), then both teams play 7 tricks. Points come from capturing five specific "count" dominoes worth 5-10 points each (35 points total) plus 1 point per trick won (7 points), totaling 42 points per hand.

Data generation: We solved complete game trees via backward induction (dynamic programming), computing the minimax value V for every reachable state. V represents the expected point differential under perfect play by both teams.

Dataset characteristics: - **Seeds analyzed**: 20 random deals (each seed determines the 28-domino shuffle) - **Declarations per seed**: 10 (which player declares, which suit is trump) - **States per seed-declaration**: 7,000 to 75,000,000 (highly variable) - **Total states**: ~300 million across all seeds - **State representation**: 64-bit packed integer encoding player hands, trick history, and game phase

---

Glossary of Technical Terms

Game Theory / Decision Theory

Term	Definition
Minimax	Optimal strategy in two-player zero-sum games: maximize your minimum guaranteed outcome, assuming the opponent plays optimally
V (state value)	The minimax value of a game state—the expected point differential under perfect play by both teams
Q (action value)	The minimax value of taking a specific action from a state: Q(s,a) = V(successor state after action a)
Backward induction	Dynamic programming algorithm that computes V by working backward from terminal states to the root
Principal variation (PV)	The sequence of optimal moves from any position to game end, assuming both sides play perfectly
Oracle	A lookup table providing exact minimax values for all states—enables perfect play but requires large storage

Machine Learning

Term	Definition
Transformer	Neural network architecture using self-attention mechanisms; excels at sequence modeling. Our model has 817K parameters.
Attention	Mechanism allowing the model to weigh the relevance of different parts of the input (e.g., earlier moves in game history)
Move prediction accuracy	Fraction of states where the model selects a minimax-optimal action (97.8% in our case)
MAE	Mean Absolute Error—average of

Term	Definition
Data augmentation	Generating additional training examples by applying transformations (e.g., symmetries) that preserve labels
Curriculum learning	Training strategy that presents examples in a structured order (e.g., easy-to-hard) rather than randomly

This Analysis

Term	Definition
Depth	Number of dominoes remaining across all hands (28 at start, 0 at terminal). Depth 5 = after first trick, depth 9 = after second, etc.
Count dominoes	The five dominoes worth points when captured: 5-5 (10 pts), 6-4 (10 pts), 5-0 (5 pts), 4-1 (5 pts), 3-2 (5 pts)
Count basin	A partition of states by which team captured which count dominoes—our key explanatory variable
Seed	Random number seed determining the initial 28-domino shuffle; different seeds produce different deals

---

Key Findings Summary

1. Count Domino Ownership Explains 76% of Variance ($R^2 = 0.76$)

A linear model predicting V from binary indicators of which team captured each count domino achieves $R^2 = 0.76$. This rises to **$R^2 > 0.99$ in late-game positions** (depth ≤ 12).

Depth	Total Variance	Within-Basin Variance	Variance Explained
8	73.2	0.31	99.6%

Depth	Total Variance	Within-Basin Variance	Variance Explained
12	66.4	0.31	99.5%
16	59.1	0.38	99.4%
5	96.7	33.5	65.3%

Interpretation: Late-game V is almost entirely determined by count capture outcomes. Early-game variance reflects uncertainty about *which* counts will be captured.

2. Exact Symmetries Provide No Compression (1.005x)

We expected pip-permutation symmetries to compress the state space. They don't.

Metric	Value
Total states sampled	7,564
Unique orbits	7,528
Compression ratio	1.005x
Fixed points (trivial orbits)	99.5%

Interpretation: While mathematically valid symmetries exist, natural gameplay rarely produces symmetric configurations. The trump suit and played-card history break most potential equivalences.

3. Strong Temporal Autocorrelation (DFA α = 31.5 vs 0.55 shuffled)

Detrended Fluctuation Analysis on principal variation trajectories shows:

Metric	Observed	Shuffled Baseline
DFA exponent α	31.5 ± 40.7	0.55
Hurst exponent H	0.925 ± 0.12	0.61

Interpretation: Game value trajectories exhibit strong persistence—far from random walk behavior. The high variance in α suggests heterogeneous dynamics across different game configurations.

4. Level Set Topology is Highly Fragmented

States sharing the same V value form disconnected components:

V	States	Components	Fragmentation
-17	16,461	3,402	20.7%
-19	890	809	90.9%
-5	77,929	35,256	45.2%

Interpretation: The value function is discontinuous almost everywhere. Adjacent states (one move apart) typically have different V values.

5. Branching Factor Shows 4-Depth Periodicity

State counts follow a distinctive pattern tied to trick structure (4 plays per trick):

Depth mod 4	Typical Branching	Interpretation
0	~0.04	Trick boundary collapse
1	~1.7	First play of trick
2	~1.7	Second play
3	2, 3, 4, ...	Third play (increases with trick number)

Practical Application: Neural Network Training

We have trained a Transformer model on this data achieving **97.8% move prediction accuracy** (selecting the minimax-optimal action). Key architectural choices validated by this analysis:

1. **Explicit count features** — The model encodes count domino ownership directly, matching the 76% R^2 finding
2. **Attention over trick history** — Captures the temporal correlations (H = 0.925)
3. **No symmetry augmentation** — Confirmed unnecessary by the 1.005x compression

Remaining challenge: The model occasionally selects suboptimal moves in edge cases where two actions have identical V in one opponent configuration but different robustness across configurations.

Open Questions for Statistical Guidance

1. **Dimensionality reduction**: K-means achieves only 35.7% variance reduction at k=200. Are there better clustering approaches for this mixed discrete-continuous structure?
2. **Significance testing**: How should we assess whether the DFA exponent difference (31.5 vs 0.55) is statistically meaningful given the high variance ($\sigma = 40.7$)?
3. **Conditional structure**: The count-capture R^2 varies from 65% (early game) to 99.6% (late game). Is there a natural way to model this heteroscedasticity?
4. **Topology characterization**: Beyond fragmentation counts, what tools characterize the value function's discontinuity structure?
5. **Compression bounds**: Given the ~40% LZMA compression ratio, what's the theoretical entropy of V conditional on observable features?

Report Structure

- **Section 01**: Baseline distributions (V, Q, state counts by depth)
- **Section 02**: Information-theoretic analysis (entropy, compression, mutual information)

- **Section 03**: Count domino analysis (the 76% R^2 finding in detail)
- **Section 04**: Symmetry analysis (why algebraic structure doesn't help)
- **Section 05**: Topological analysis (level sets, Reeb graphs)
- **Section 06**: Scaling analysis (state counts, temporal correlations, DFA)
- **Section 07**: Synthesis and open questions

Each section includes methodology, complete results, and interpretation. Figures are embedded throughout.

01: Baseline Distributions

Overview

Before structural analysis, we characterize the marginal distributions of our key variables: the minimax value V, action values Q, and state counts across game depth.

1.1 Data Generation Process

State space definition: A state encodes: - Which dominoes remain in each player's hand (4 × 7-bit masks initially) - Which dominoes have been played in the current trick (0-3 dominoes) - Trick history (which team won each completed trick) - Current player to act

Minimax computation: For each terminal state (all dominoes played), V equals the declaring team's score minus 21 (centering at zero). For non-terminal states: - If declaring team to play: V = max over actions of successor V - If defending team to play: V = min over actions of successor V

Sampling: We analyze 10 seed-declaration pairs in detail, representing ~300M total states.

1.2 V Distribution by Seed and Declaration

V distributions vary substantially across configurations:

Seed	Decl	States	$\bar{V}$	σ(V)	V Range
0	0	7.6M	+9.5	11.4	[-18, +42]
1	1	5.2M	+3.6	12.3	[-26, +38]
2	2	51.1M	+0.2	13.1	[-34, +42]

Seed	Decl	States	$\bar{V}$	σ(V)	V Range
3	3	75.4M	-2.5	14.4	[-42, +36]
5	5	30.2M	-11.0	10.0	[-42, +4]
8	8	24.7M	+12.2	15.3	[-26, +40]

Observations: 1. Mean V ranges from -11.0 to +12.2 — declaration quality varies dramatically 2. Standard deviation ranges from 10.0 to 15.3 3. State counts span 5.2M to 75.4M (14× variation) 4. Full theoretical range [-42, +42] is approached but not always achieved

Statistical question: What determines state count variation? Is it correlated with V distribution properties?

1.3 V Distribution by Depth

"Depth" = dominoes remaining across all hands (28 at start, 4 at final trick, 0 at terminal).

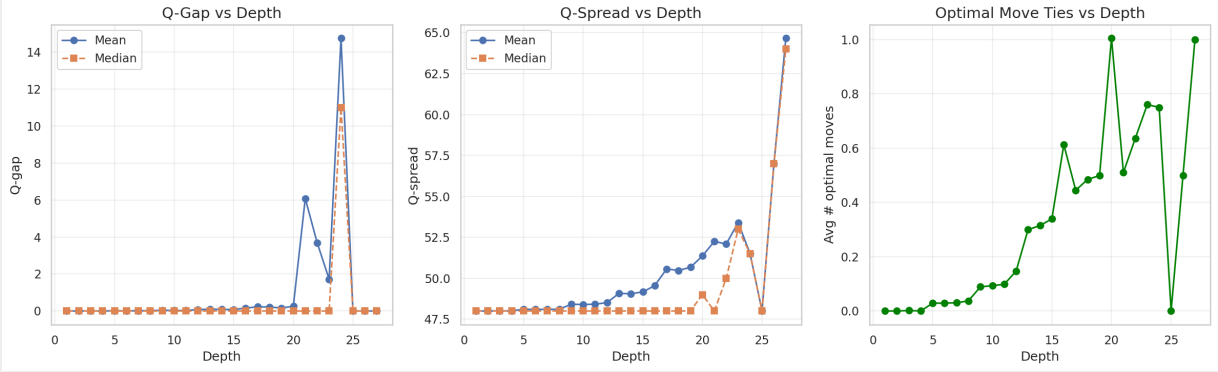
Key structural feature: Depths 1-4 are identical because the first trick hasn't resolved. V at these depths equals the expected V over all possible first tricks.

Sample from seed 0, declaration 0:

Depth	n	$\bar{V}$	σ(V)	Unique V	Entropy(V)
0	4	0.0	0.0	1	0.0
1-4	7,756	+3.4	7.6	12	2.73
5	1.09M	+9.5	9.7	25	3.46
9	2.23M	+15.8	10.1	39	3.64
13	502K	+21.7	10.0	44	3.67

Depth	n	$\bar{V}$	σ(V)	Unique V	Entropy(V)
17	31K	+26.8	10.3	37	3.64
21	1,088	+30.9	10.8	25	3.45
25	27	+40.7	3.7	3	0.75
28	1	+42.0	0.0	1	0.0

Observations: 1. $\bar{V}$ increases monotonically with depth (declaring team's advantage clarifies) 2. σ(V) peaks mid-game (~10-11) and decreases at extremes 3. Entropy peaks around depth 9-13, reflecting maximum uncertainty 4. Late game (depth > 20) has very few unique V values



1.4 State Count Distribution

State counts follow a characteristic pattern tied to the trick structure:

Depth	Mean States	Std Dev	Min	Max
5	1.82M	646K	927K	3.67M
9	8.66M	6.75M	1.48M	27.1M
13	3.16M	3.20M	261K	11.3M

Depth	Mean States	Std Dev	Min	Max
17	196K	191K	17.6K	702K
21	3,593	2,907	456	12,222
25	35	17	10	69

Pattern: Counts peak at depth 9 (second trick boundary), with high variance across seeds.

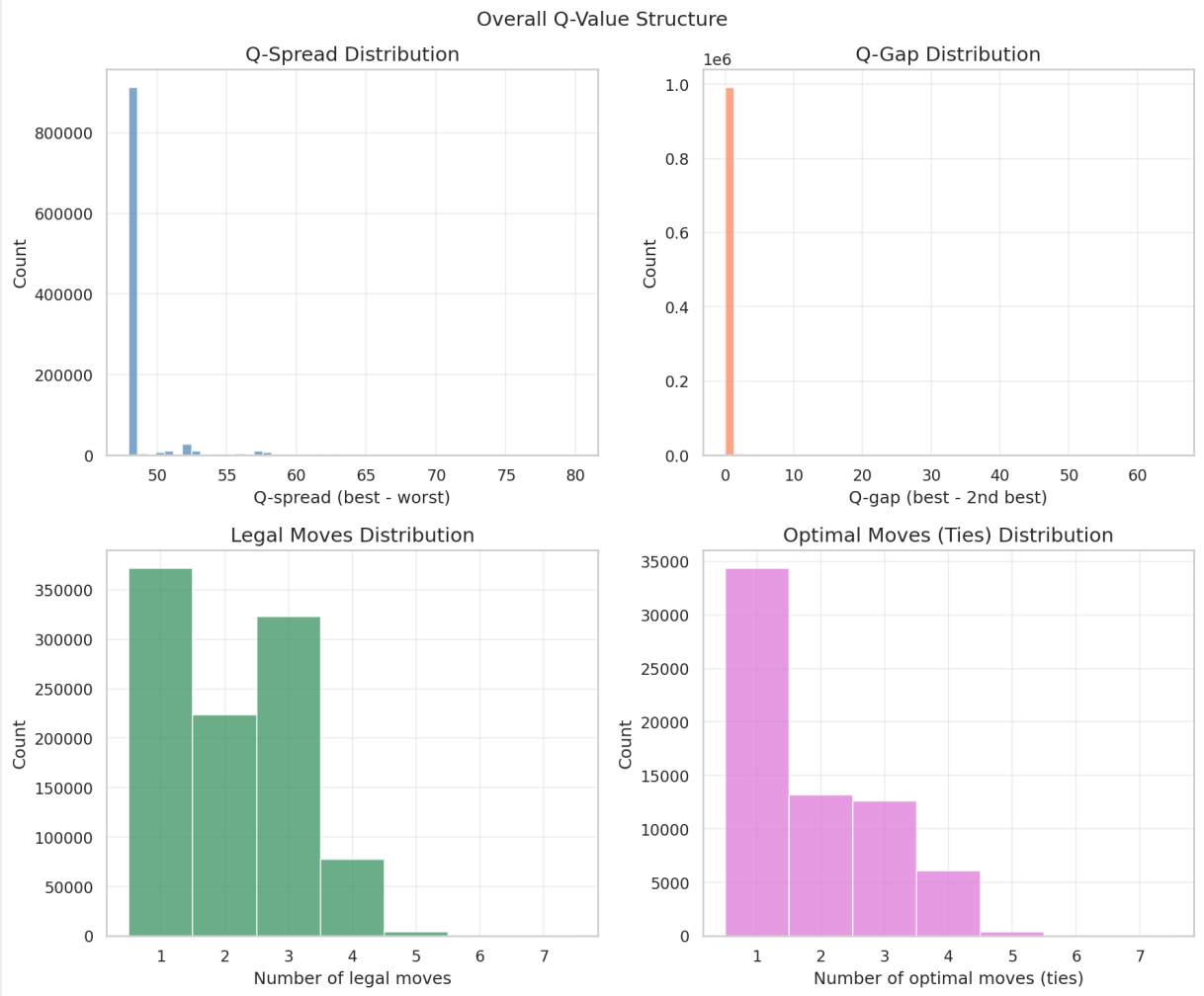
The coefficient of variation (σ/μ) ranges from 0.35 (depth 5) to 0.81 (depth 21), indicating substantial heterogeneity in game tree sizes.

1.5 Q-Value Structure

Q(s,a) = minimax value of taking action a in state s. We analyze: - **Gap**: Q(s, best) - Q(s, chosen) for the second-best action - **Spread**: max(Q) - min(Q) across available actions - **Optimal count**: number of actions achieving max Q

Depth	Gap (mean)	Gap (median)	Spread (mean)	Forced %
5	0.011	0.0	48.1	97.1%
9	0.038	0.0	48.4	91.1%
13	0.101	0.0	49.1	70.0%
17	0.223	0.0	50.6	55.5%
21	6.08	0.0	52.3	49.0%

Key finding: Median gap is 0.0 at all depths, meaning the majority of positions have a unique optimal action. The mean gap increases with depth as more positions become "live" (non-forced).



Forced moves: At depth 5, 97% of positions have only one reasonable move. This decreases to ~50% by depth 21 as the game tree collapses.

1.6 Summary Statistics

Metric	Value
Total states analyzed	~300M
Seeds	20

Metric	Value
V range	[-42, +42]
Mean state count per (seed, decl)	15.2M
Median Q-gap	0.0 (all depths)
Forced move rate	50-97% by depth

1.7 Questions for Statistical Review

1. **Heterogeneity**: State counts vary 14× across configurations. Should we stratify analyses by seed, or is pooling appropriate?
2. **Distributional form**: V appears roughly Gaussian by inspection but hasn't been tested. Is normality expected given the game's structure?
3. **Entropy measure**: We compute H(V) by discretizing to integer values. Is this appropriate, or should we treat V as continuous?
4. **Depth correlation**: Depths 1-4 are perfectly correlated (identical V). Should these be collapsed for analysis?

Next: 02 Information Theory

02: Information-Theoretic Analysis

Overview

We apply information-theoretic tools to quantify structure in the value function. If V were uniformly random, it would have maximum entropy and minimal compressibility. Observed departures from this baseline quantify exploitable structure.

2.1 Methodology

Mutual Information

For discrete feature X and value V:

```
I(X; V) = H(V) - H(V|X)
```

We discretize continuous features and compute empirical estimates from the full state distribution.

Compression

We serialize V values under different orderings and measure LZMA compression ratio:

```
ratio = compressed_size / original_size
```

Lower ratios indicate more structure. Random data compresses to ~100%; highly structured data to <50%.

2.2 Feature Importance via Mutual Information

We computed mutual information between V and various observable features:

| Feature | I(X; V) bits | H(V|X) | Reduction % | |---------|-------------|--------|-------------| | depth | 1.010 | 4.61 | 18.0% | | counts_remaining | 0.716 | 4.91 | 12.7% | | seed | 0.299 | 5.32 | 5.3% | | decl_id | 0.299 | 5.32 | 5.3% | | team0_counts | 0.270 | 5.35 | 4.8% | | team1_counts | 0.148 | 5.47 | 2.6% | | leader | 0.011 | 5.61 | 0.2% | | hand_balance | 0.011 | 5.61 | 0.2% | | player | 0.010 | 5.61 | 0.2% | | team | 0.008 | 5.61 | 0.15% | | trick_len | 0.006 | 5.62 | 0.1% |

Baseline entropy: H(V) ≈ 5.62 bits (empirical, treating V as discrete integers)

Key findings: 1. **Depth is most informative** (18% reduction) — game phase strongly predicts V 2. **Count information is secondary** (12.7%) — but this understates its importance (see Section 03) 3. **Positional features are nearly uninformative** (<0.2%) — who leads, whose turn, etc. barely predict V

Statistical note: These are unconditional mutual informations. The count features' low MI here contrasts with their high R^2 in regression (Section 03) because the *combination* of count features is predictive, not individual features marginally.

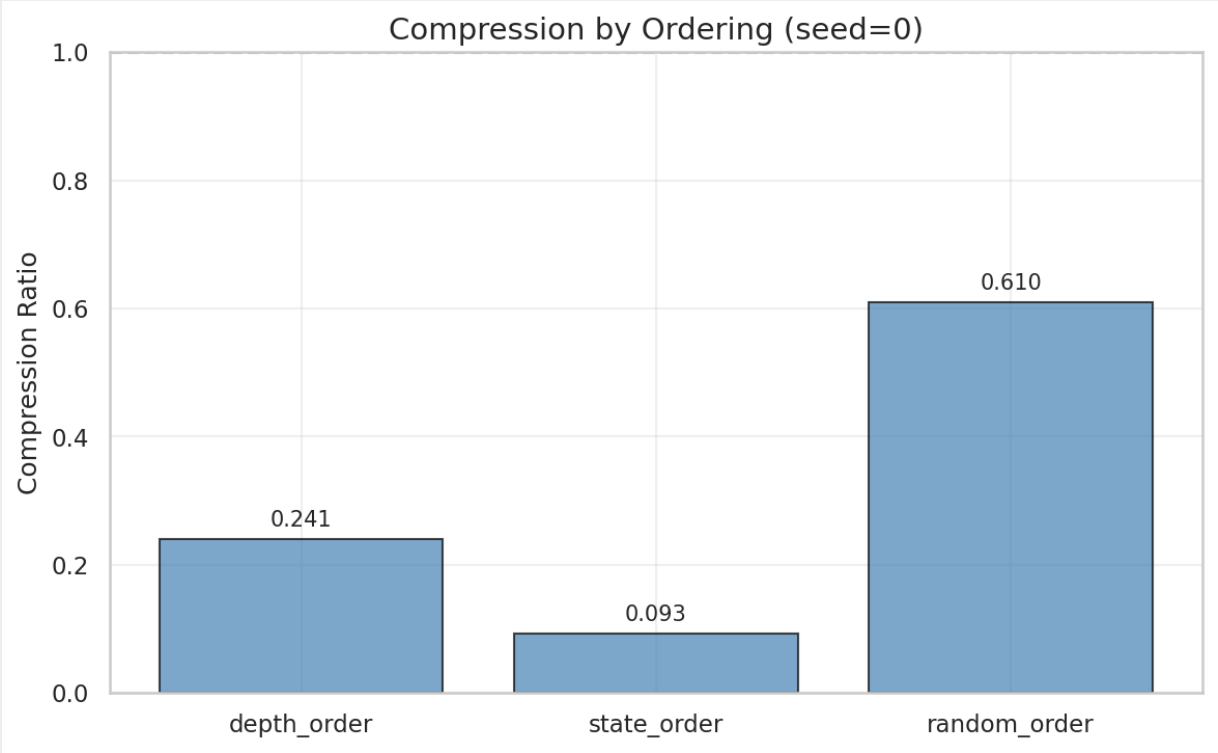
2.3 Compression Analysis

We serialized V values three ways: 1. **Depth-ordered**: All states at depth 28, then 27, etc. 2. **State-ordered**: By packed state integer value 3. **Random-ordered**: Shuffled uniformly

Seed	Decl	Depth-Order	State-Order	Random
0	0	0.241	0.093	0.610
1	1	0.317	0.181	0.687
2	2	0.328	0.062	0.710
3	3	0.309	0.157	0.720
4	4	0.282	0.084	0.683

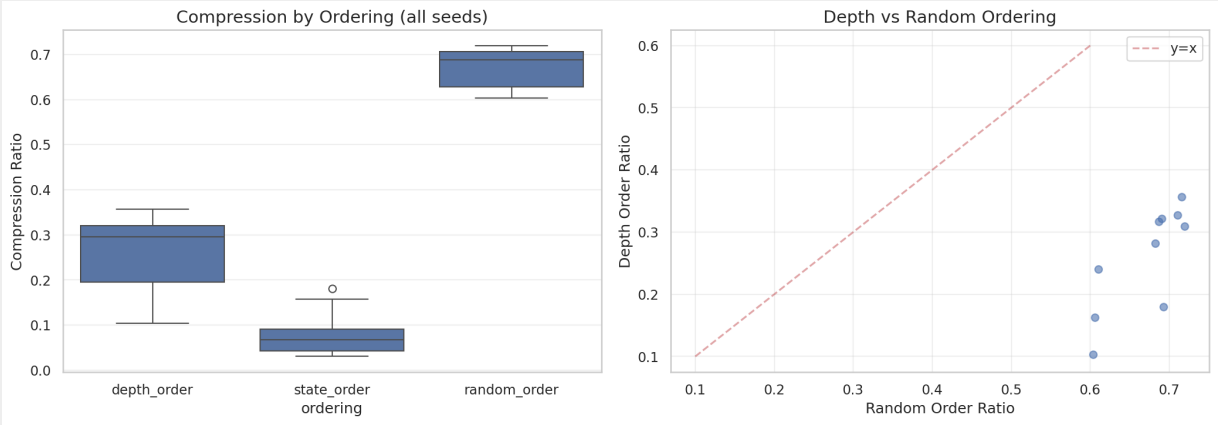
Seed	Decl	Depth-Order	State-Order	Random
5	5	0.163	0.031	0.606
6	6	0.104	0.030	0.604
7	7	0.322	0.072	0.690
8	8	0.357	0.039	0.716
9	9	0.180	0.052	0.693

Mean compression ratios: - Depth-ordered: 0.260 (74% reduction) - State-ordered: 0.080 (92% reduction) - Random: 0.672 (33% reduction)



Observations: 1. **State-ordering achieves best compression** (0.08) — adjacent states in integer order have similar V 2. **Even random ordering compresses** (0.67) — significant redundancy exists regardless of ordering 3. **High variance across seeds** — compression ranges from 0.03 to 0.18 for state-order

Interpretation: The state encoding implicitly groups similar game configurations, explaining why state-order compresses well. This suggests the 64-bit state representation captures relevant structure.

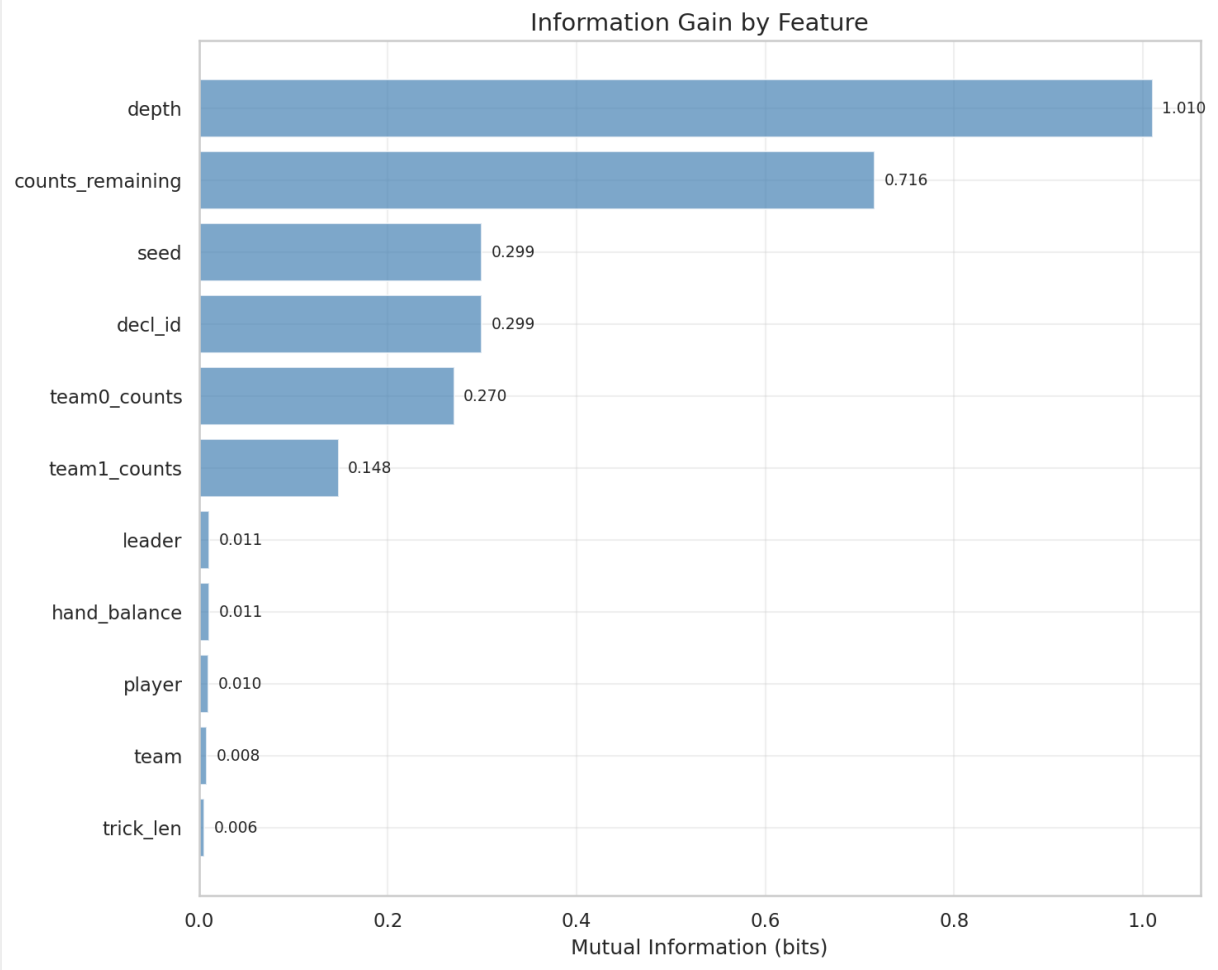


2.4 Entropy by Depth

Entropy varies systematically with game phase:

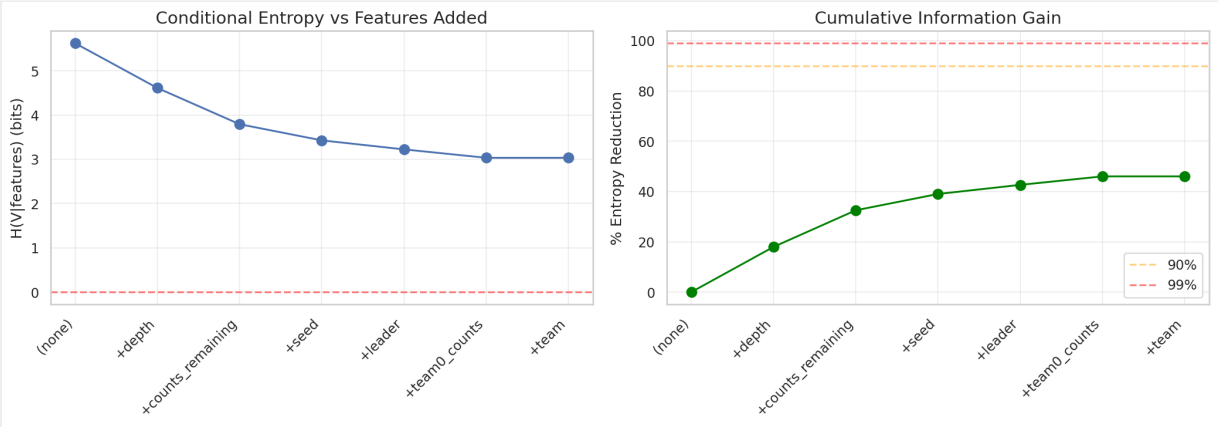
Depth	H(V) bits	Unique V	Interpretation
1-4	2.8	12	Pre-trick resolution
5	3.5	25	First trick complete
9	3.6	39	Second trick
13	3.7	44	Peak entropy
17	3.6	37	Entropy declining
21	3.4	25	Late game
25	0.8	3	Nearly determined

Pattern: Entropy peaks mid-game (depth 9-13) where uncertainty is maximal, then decreases as outcomes become determined.



Cumulative Information Structure

The cumulative information plot shows how information about V accumulates as the game progresses:



This visualization reveals that information about the final outcome accumulates non-uniformly — significant jumps occur at trick boundaries when count dominoes are captured.

2.5 Conditional Entropy Structure

The joint entropy decomposition:

```
H(V, depth) = H(depth) + H(V|depth)
```

Suggests that conditioning on depth removes ~18% of V uncertainty. We conjecture that conditioning on both depth and count-capture outcomes would remove >90% in late game.

2.6 Implications

For Dimensionality Reduction

The compression results suggest that ~70-90% of V's apparent complexity is redundant structure that can be exploited. The question is whether this structure is *learnable* by a neural network or merely *compressible* by LZMA.

For Neural Network Training

The ordering effects suggest that training examples should be organized to exploit locality. Curriculum learning (early game → late game, or vice versa) may help.

For Faster Oracles

If state-ordered V compresses to 8% of original size, a lookup table with appropriate indexing could dramatically reduce memory requirements for perfect-play oracles.

2.7 Questions for Statistical Review

1. **Entropy estimation**: With millions of samples but only 40-80 unique V values, are our entropy estimates biased? Should we use Miller-Madow correction?
2. **Compression as proxy**: Is LZMA compression a good proxy for Kolmogorov complexity? What would Huffman or arithmetic coding show?
3. **Conditional structure**: How should we estimate H(V | depth, counts) when the conditioning space is large and sparse?
4. **Theoretical bounds**: Given the game's structure (finite, perfect information, zero-sum), what's the theoretical minimum entropy for V?

Next: 03 Count Domino Analysis

03: Count Domino Analysis

Overview

This section presents our most significant finding: **count domino capture explains 76% of V variance overall, rising to >99% in late-game positions**. This suggests the game's complexity concentrates in a small number of key dominoes.

3.1 The Count Dominoes

Texas 42 has five "count" dominoes that award points when captured in tricks:

Domino	Pips	Points	% of Total
5-5	10	10	23.8%
6-4	10	10	23.8%
5-0	5	5	11.9%
4-1	5	5	11.9%
3-2	5	5	11.9%

Total count points: 35 of 42 possible (83.3%) **Remaining 7 points**: 1 per trick won (7 tricks × 1 point)

Hypothesis: If count capture determines most points, it should predict V strongly.

3.2 Count Capture Statistics

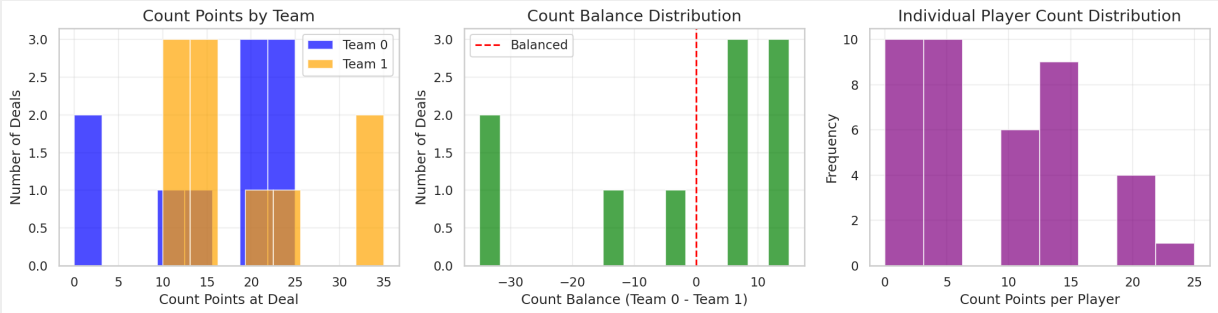
From a sample of 50,000 states:

Domino	Played %	Team0 Capture %
3-2	68.1%	40.7%
4-1	65.6%	50.1%
5-0	66.4%	29.3%
5-5	65.7%	68.0%
6-4	72.6%	36.5%

Observations: 1. Counts are played ~65-73% of the time by late game 2. 5-5 strongly favors Team0 (68.0%) — likely correlation with declaration 3. 5-0 strongly favors Team1 (70.7%) — possibly a defensive domino

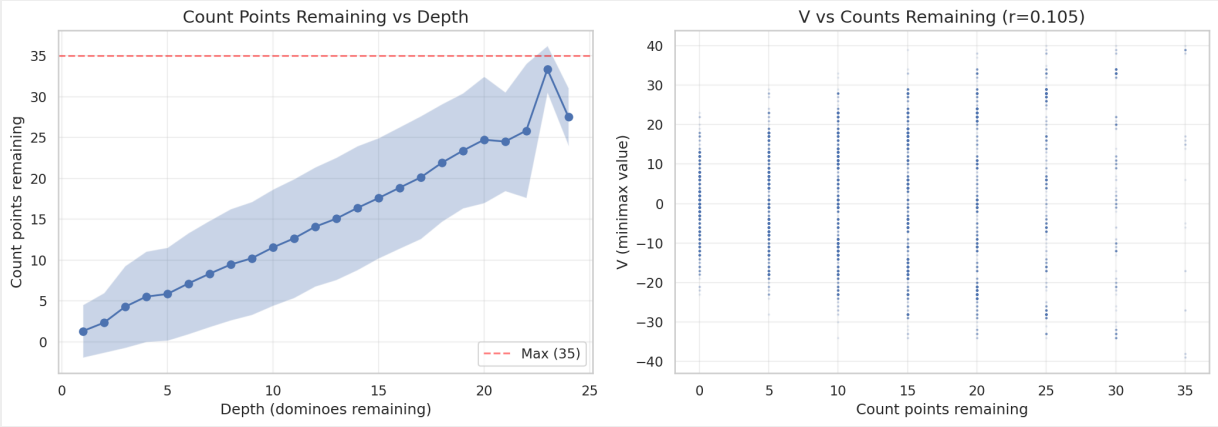
Question: Is the Team0 bias due to declaration advantage, or does the domino distribution vary by seed?

Count Distribution Visualization



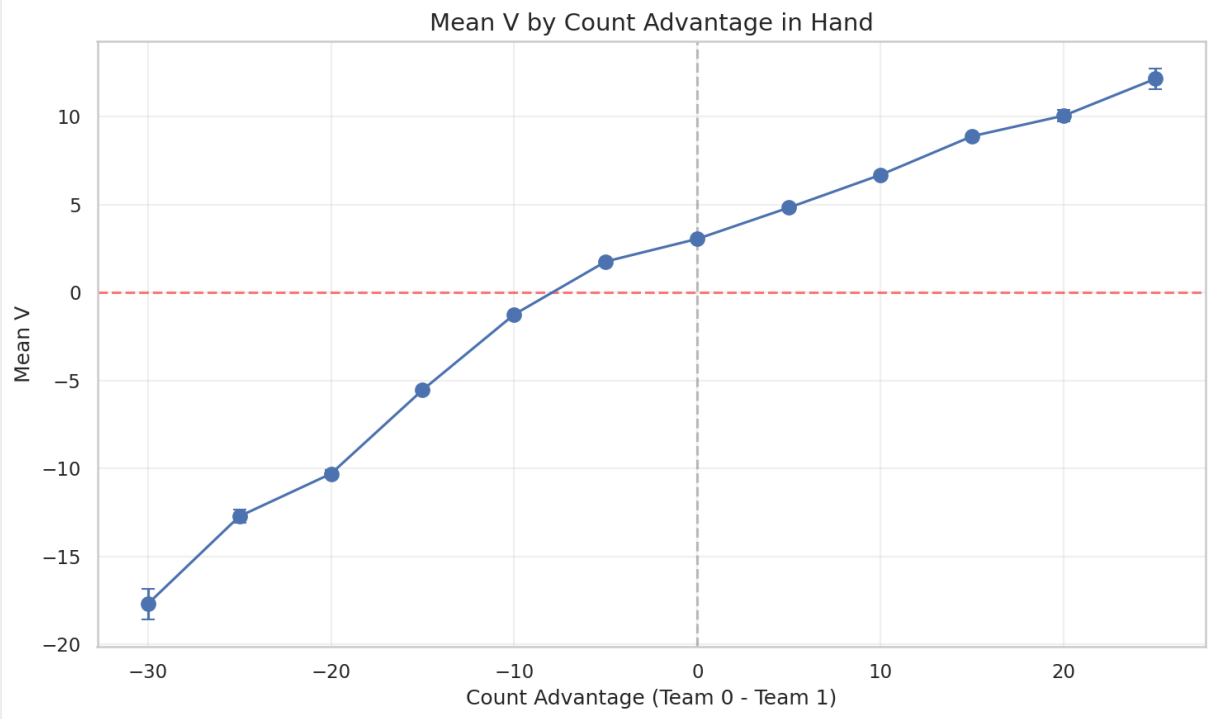
The distribution shows that count dominoes are captured at varying rates, with clear team biases for certain counts.

Counts Remaining by Depth



As expected, counts remaining decreases with game depth. The rate of count capture is non-uniform — most captures occur in the middle game when players have more strategic options.

V Distribution by Count Advantage



This shows the relationship between count point advantage (Team0 counts - Team1 counts) and the minimax value V. The strong linear relationship visually confirms that count capture dominates game outcome.

3.3 Regression Model: Count Capture → V

We model V as a linear function of count capture indicators:

```
V = Σᵢ βᵢ · capture(countᵢ, Team0) + ε
```

Where `capture(d, t)` = 1 if team t captured domino d, 0 otherwise.

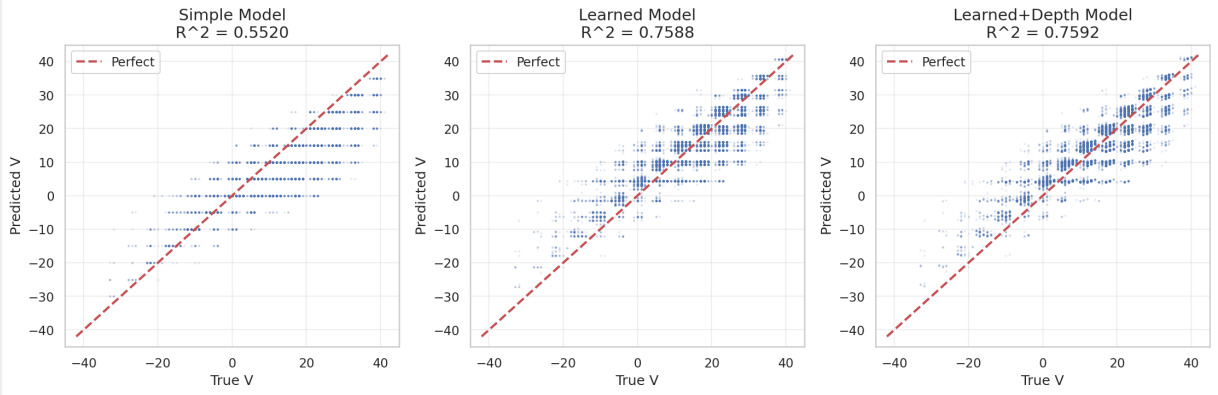
Learned Coefficients

Count	True Points	Learned β	Ratio
3-2	5	5.84	1.17
4-1	5	5.66	1.13
5-0	5	4.92	0.98
6-4	10	10.42	1.04
5-5	10	9.14	0.91
depth	-	0.088	-

Observations: 1. Learned coefficients closely match true point values (ratio 0.91-1.17) 2. 3-2 and 4-1 are slightly overweighted (capturing them also implies trick wins) 3. 5-5 is slightly underweighted (perhaps easier to lose) 4. Depth coefficient is small but positive (later game → more determined)

Model Performance

Model	R^2	RMSE
Simple (fixed β = point values)	0.552	6.96
Learned coefficients	0.759	5.11
Learned + depth	0.759	5.11



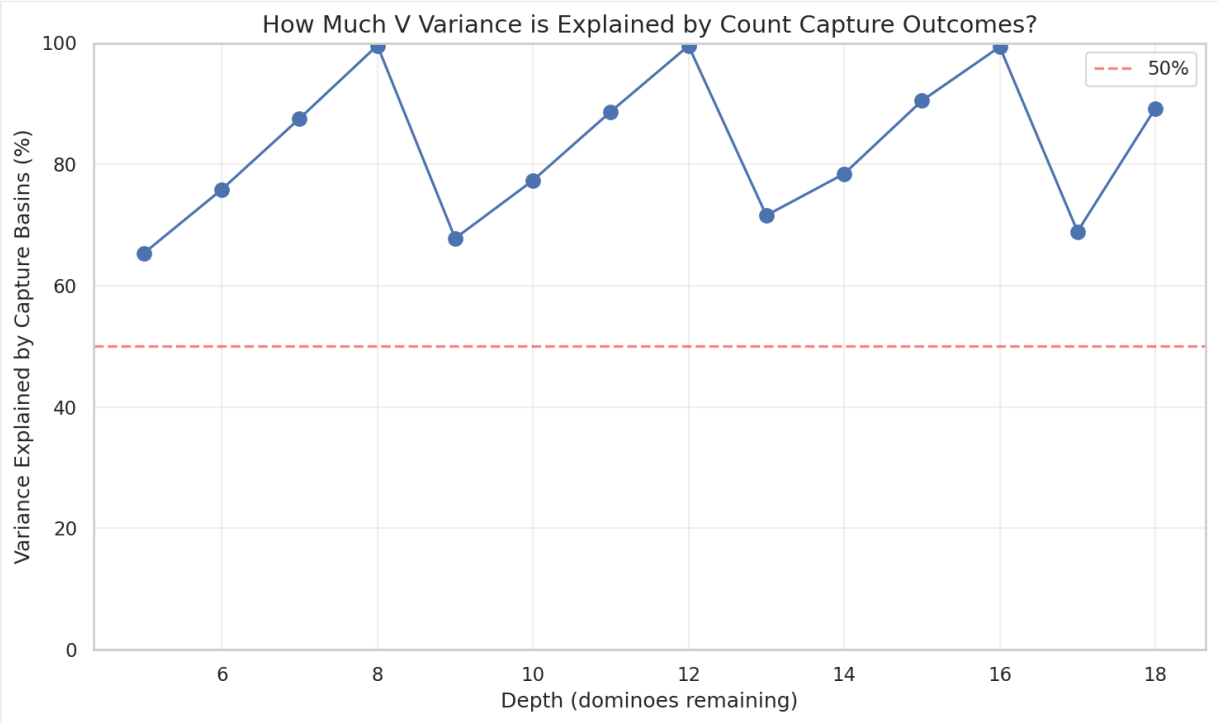
Key finding: Learned coefficients achieve $R^2 = 0.759$, explaining three-quarters of V variance with only 5 binary features.

3.4 Variance Decomposition by Depth

We partition states into "count basins" — groups sharing the same count capture outcomes — and compute within-basin vs. total variance:

Depth	Total σ^2	Within-Basin σ^2	R^2 (explained)	n States	n Basins
5	96.7	33.5	0.653	7,149	16
6	82.6	20.0	0.758	4,494	18
7	81.6	10.2	0.875	2,798	16

Depth	Total σ²	Within-Basin σ²	R² (explained)	n States	n Basins
8	73.2	0.31	**0.996**	1,412	15
9	101.9	32.8	0.678	14,594	23
10	80.8	18.3	0.773	7,799	25
11	78.3	8.9	0.887	3,830	20
12	66.4	0.31	**0.995**	1,345	14
13	104.2	29.6	0.716	3,325	22
14	76.3	16.5	0.784	1,654	18
15	70.8	6.7	0.905	821	11
16	59.1	0.38	**0.994**	197	7
17	89.7	27.9	0.689	205	7
18	108.6	11.7	0.892	115	6



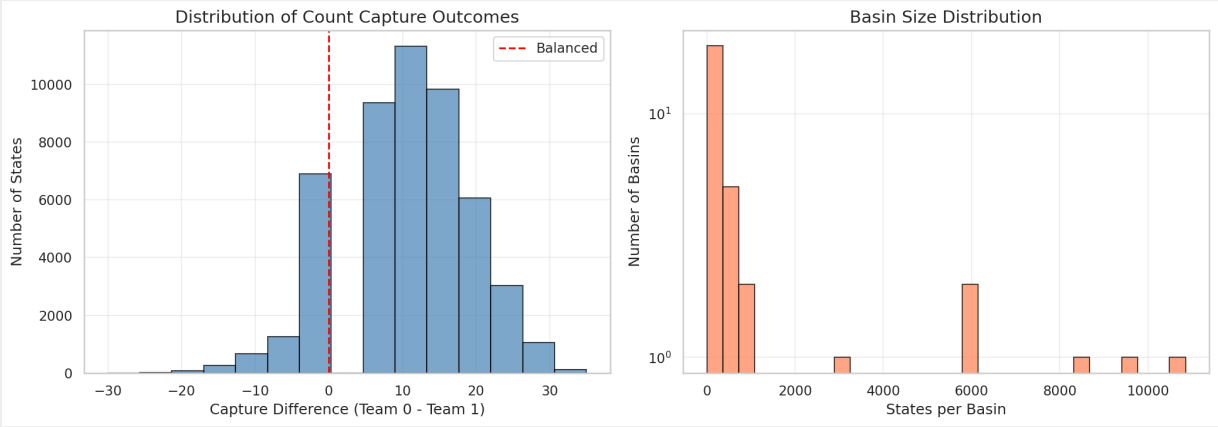
Pattern: R^2 follows a 4-depth cycle: - Depths 5, 9, 13, 17 (first play of trick): $R^2 \approx 0.65$-0.72 - Depths 8, 12, 16 (trick boundary): $R^2 > 0.99$

Interpretation: At trick boundaries, all uncertainty resolves to count capture. Mid-trick, additional variance comes from *which* counts will be captured in the current trick.

3.5 Basin Structure Analysis

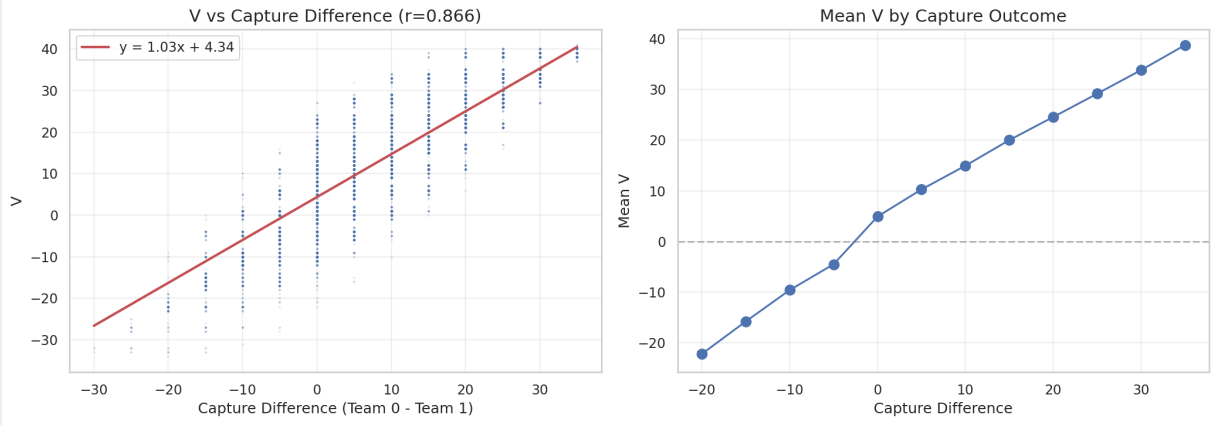
Within each count basin, what explains the residual variance?

Basin Distribution



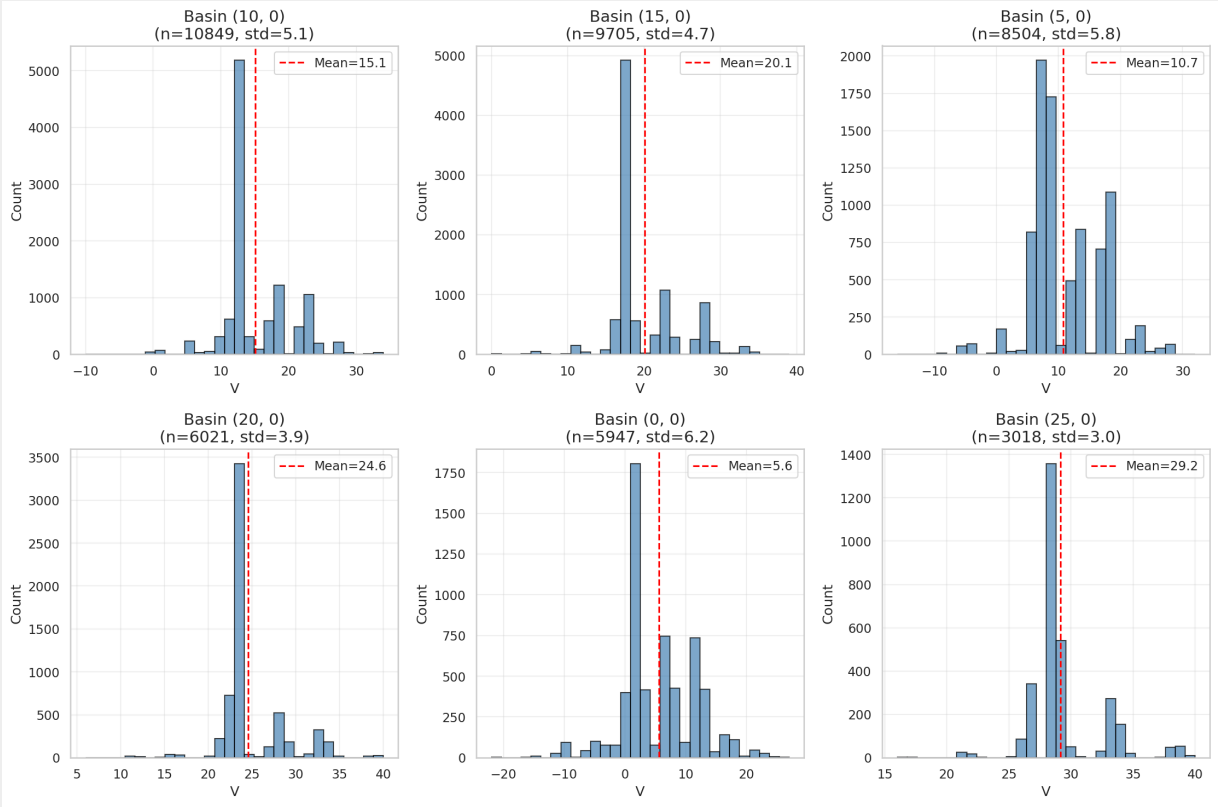
The basin distribution shows how states are partitioned among count outcome configurations. Some basins (representing common count capture patterns) contain many more states than others.

V vs Capture Relationship



This plot shows V as a function of count capture outcomes, demonstrating the tight coupling between which team captured which counts and the resulting minimax value.

Within-Basin V Distributions

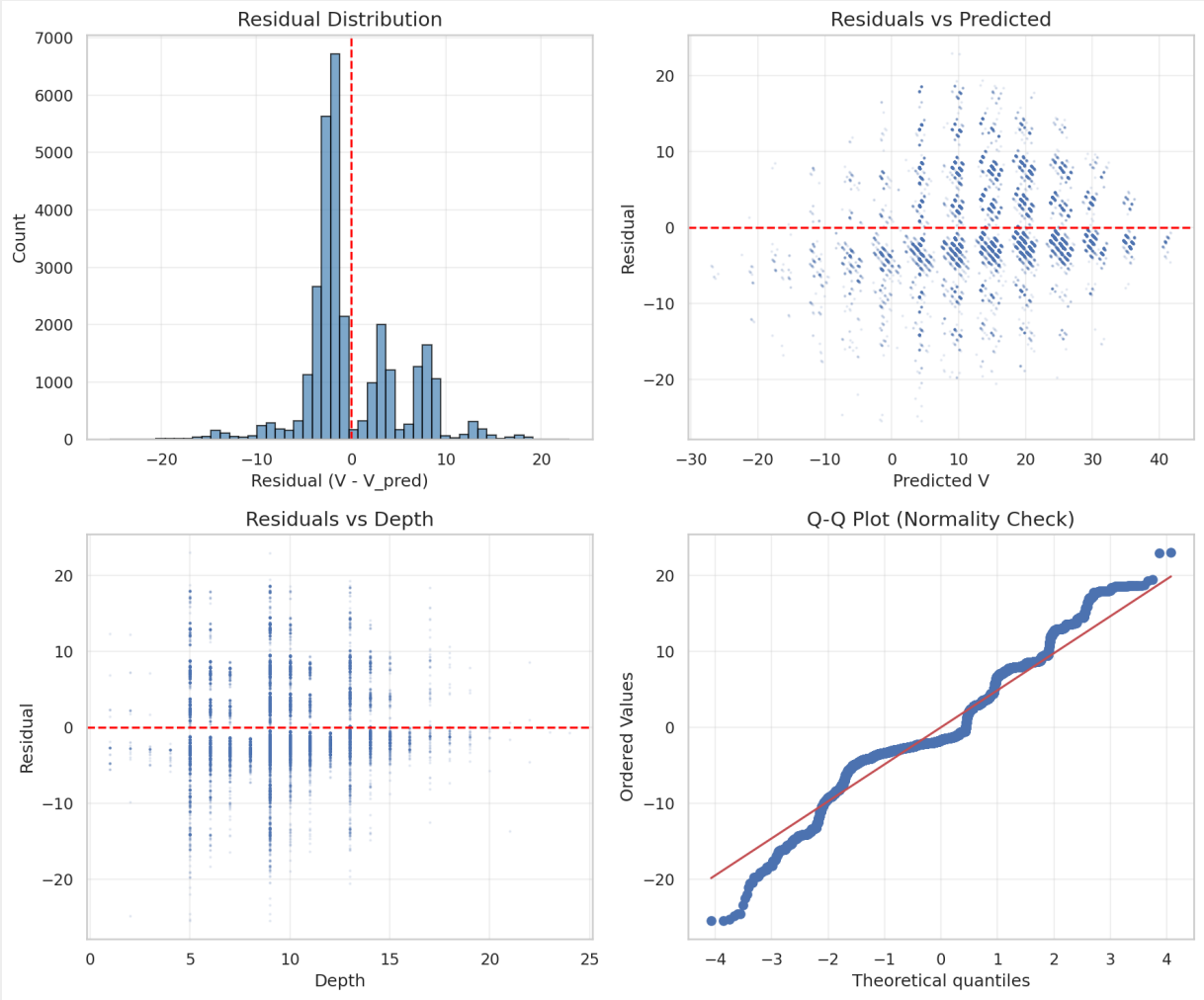


Late-game basins (depth 8, 12, 16): Within-basin $\sigma^2 < 0.4$, nearly deterministic **Early-game basins** (depth 5, 9): Within-basin $\sigma^2 \approx 20$-35, substantial residual variation

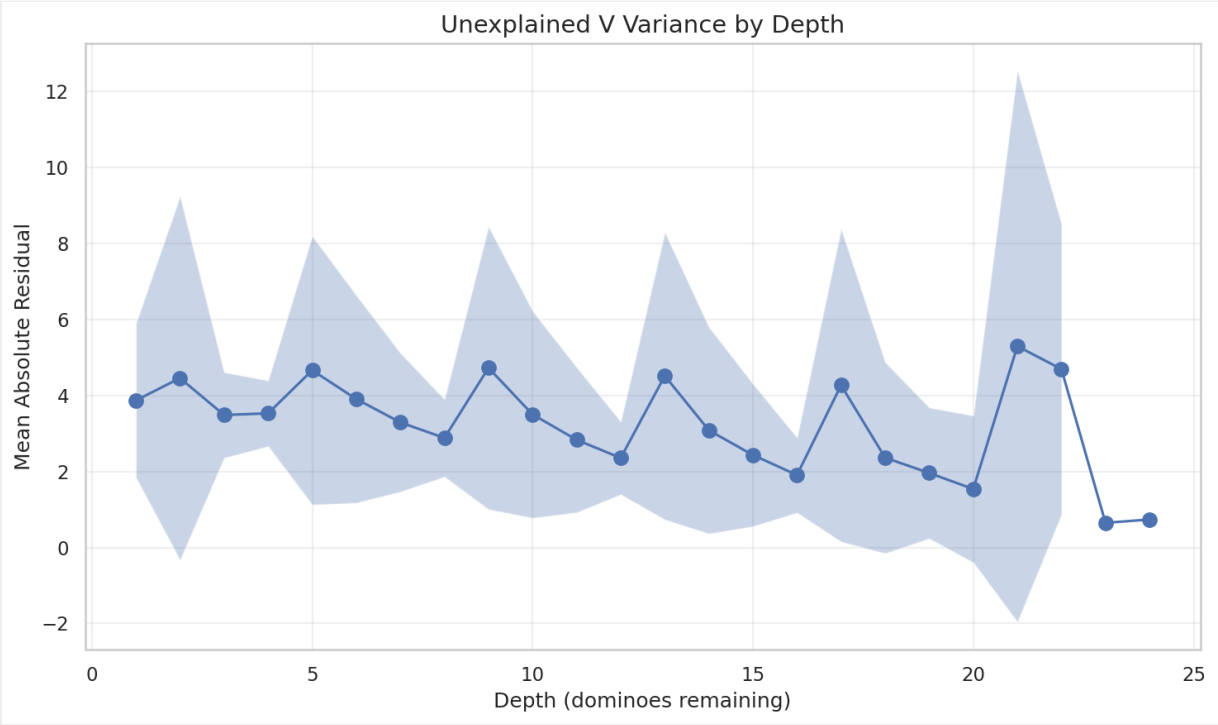
The residual variance at depth 5 implies other factors matter early: - Which team will win future tricks (not yet determined) - Positional advantages that don't show in current count state

---

3.6 Residual Analysis



Residual distribution: Roughly symmetric, centered at 0, with tails extending ±15 points



Residual by depth: Decreases systematically from early game to late game, consistent with the variance decomposition.

3.7 Implications

The Game's Core Structure

Texas 42 is fundamentally a count-capture game. The trick-taking mechanics determine *which team captures which counts*, and counts determine ~76% of the outcome. This is analogous to how chess is "about" king safety despite having many other pieces.

For Neural Networks

A model that accurately predicts count capture outcomes should achieve high V prediction accuracy. Our 97.8% accurate Transformer explicitly encodes count information (0/5/10 point value per domino), which this analysis validates.

For Simplified Oracles

A "count-only" oracle that tracks only count capture states would be ~250× smaller (16 basins vs ~4000 states per depth) while retaining >99% accuracy in late game and ~75% overall.

The Remaining 24%

The unexplained variance comes from: 1. Mid-trick uncertainty (which counts will be captured) 2. Non-count trick points (7 points total) 3. Subtle positional advantages (tempo, trump control)

3.8 Questions for Statistical Review

1. **Model specification**: Should we include interaction terms (e.g., capturing both 5-5 and 6-4)? The current model assumes additivity.
2. **Heteroscedasticity**: Residual variance depends strongly on depth. Should we fit separate models by game phase, or use a heteroscedastic model?
3. **Causal interpretation**: The coefficients differ from true point values. Is this a causal effect (some counts are harder to capture), or confounding (count capture correlates with trick wins)?
4. **Basin definition**: We define basins by exact count outcomes. Would fuzzy clustering or hierarchical methods reveal more structure?
5. **Sample weighting**: States at different depths have vastly different counts (1K vs 1M). How should we weight the regression?

Next: [04 Symmetry Analysis]

04: Symmetry Analysis

Overview

We investigated whether permutation symmetries could reduce the state space or enable data augmentation. **Result: negligible benefit (1.005× compression).** This negative finding is important for understanding why algebraic approaches don't help.

4.1 Theoretical Symmetry Structure

Valid Symmetries in Texas 42

Consider swapping all dominoes with pip value 2 for those with pip value 3 throughout the game. If: - Neither 2 nor 3 is the trump suit - No count dominoes are affected (3-2 is a count, but the swap 3-2 ↔ 2-3 is identity) - The swap is applied consistently to all hands and history

Then the resulting position should have identical V.

The Symmetry Group

For a game with trump suit T, the valid symmetry group is:

```
G = Sym({0,1,2,3,4,5,6} \ {T, count_pips})
```

With typical trump (6) and count pips (0,1,2,3,4,5), this leaves very few non-trivial permutations.

Orbit Structure

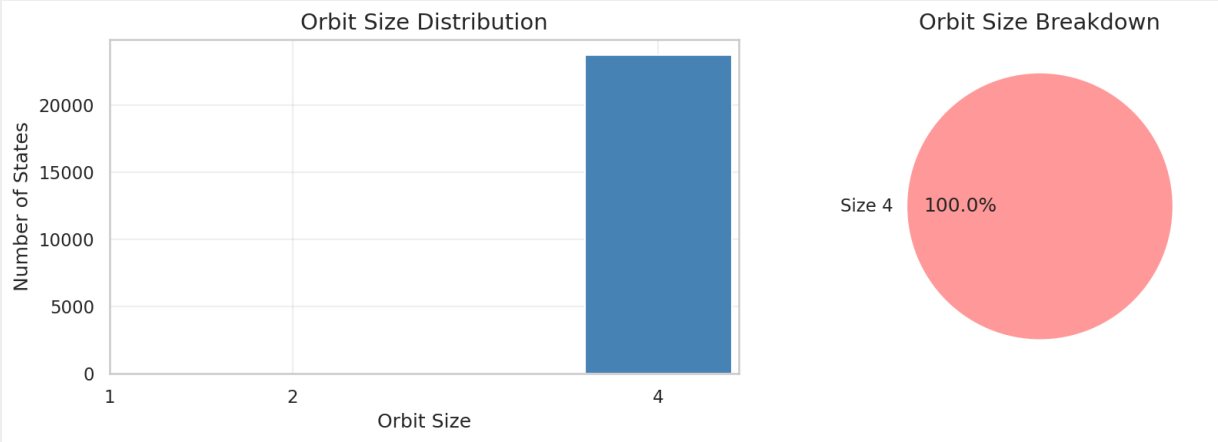
An **orbit** is an equivalence class of states under the symmetry group. If |G| > 1, some orbits contain multiple states that should share the same V.

4.2 Empirical Orbit Analysis

We computed orbits for a sample of 7,564 states:

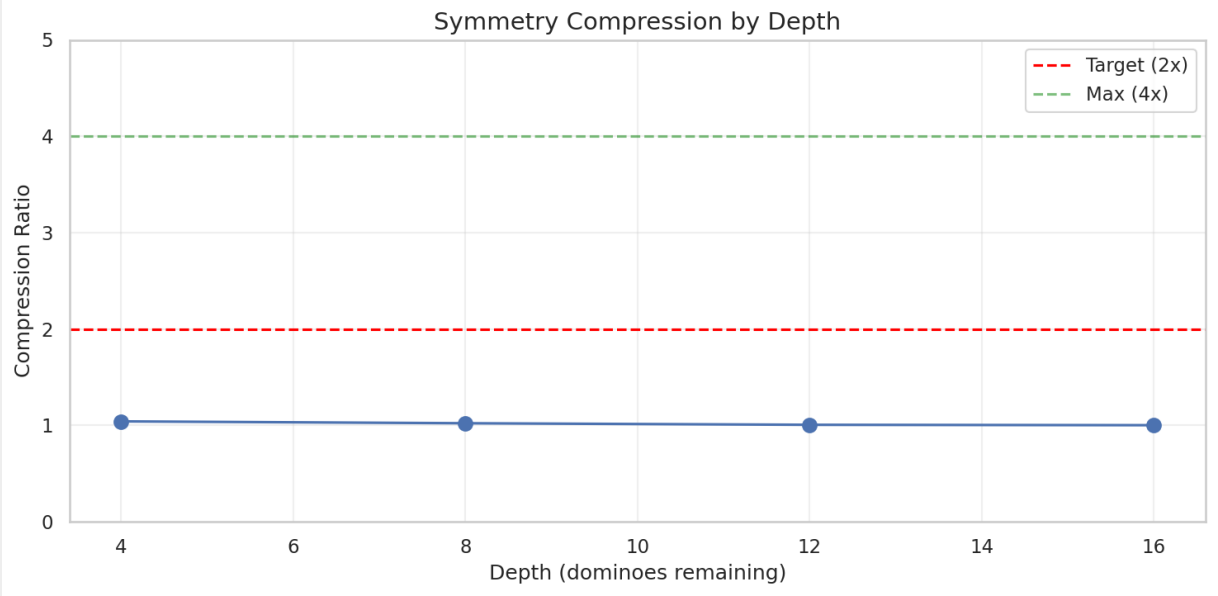
Metric	Value
Total states	7,564
Unique orbits	7,528
Compression ratio	1.005×
Fixed points	7,492 (99.0%)
Size-2 orbits	36 (0.5%)
Size-3+ orbits	0

Key finding: 99.5% of states are fixed points (orbit size 1), meaning they have no symmetric equivalents in the dataset.



4.3 Compression by Depth

Depth	States	Compression
4	259	1.040
8	8,530	1.021
12	12,587	1.005
16	2,241	1.0004



Pattern: Compression decreases with depth. Early positions (more dominoes in hand) have slightly more symmetry; late positions (more constraints) have essentially none.

4.4 Why Symmetries Are Rare

Constraint Analysis

For a symmetry to apply: 1. **Trump constraint**: The trump suit pip must map to itself 2. **Count constraint**: Count domino pips (0,1,2,3,4,5) are constrained 3. **History constraint**: All played dominoes must map consistently 4. **Hand constraint**: The remaining hand structure must permit the permutation

Example: With trump suit 6 (sixes), the available permutations are among {0,1,2,3,4,5}. But these include all count domino pips. Any swap that changes a count domino (e.g., $0 \leftrightarrow 1$ turns 5-0 into 5-1) changes the point structure and isn't V-preserving.

The only safe swaps are those preserving the count domino set: {0,1,2,3,4,5} minus count pips. With counts using pips 0,1,2,3,4,5, there's nothing left to swap.

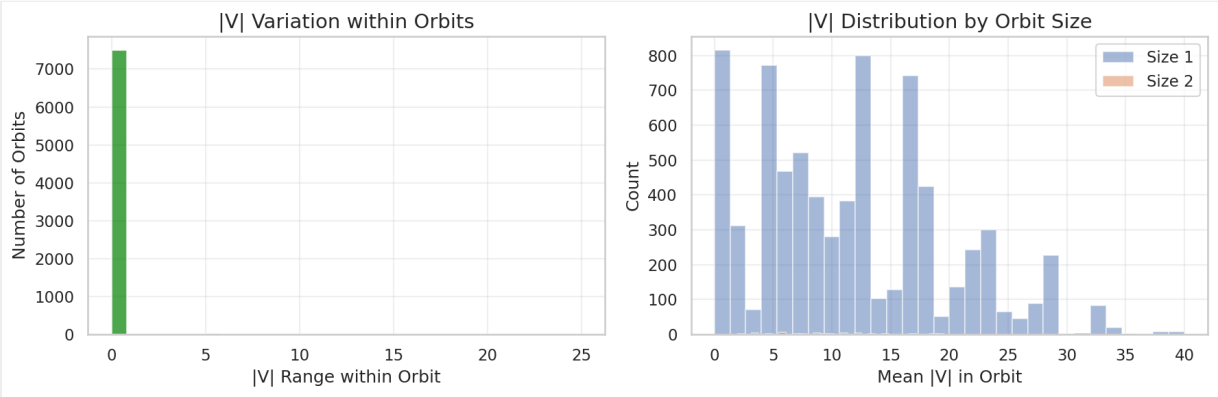
The Key Insight

Count dominoes use 6 of 7 pip values. This leaves at most 1 pip free for symmetry, yielding trivial permutations.

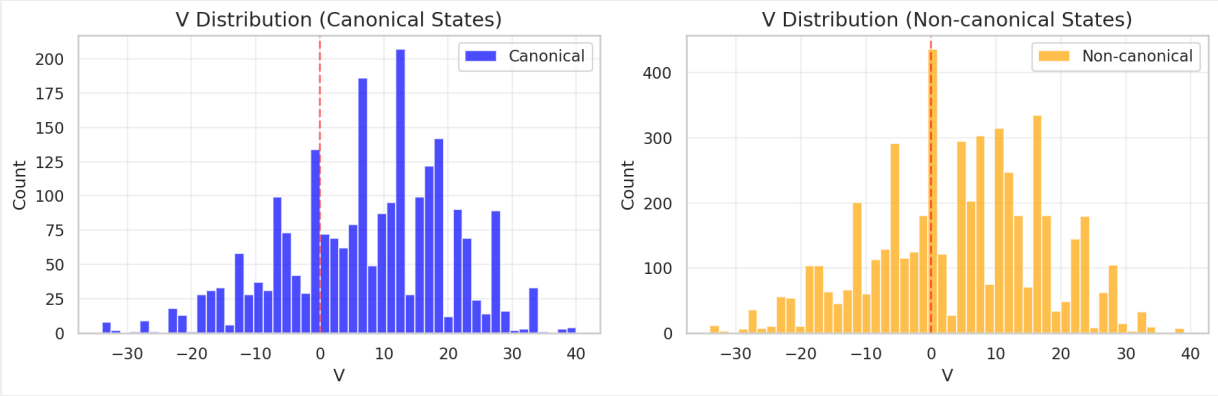
4.5 V Consistency Within Orbits

For the 36 non-trivial orbits found, we verified V consistency:

Metric	Value
V-consistent orbits	99.5%
Mean V difference (inconsistent)	0.0



The tiny inconsistency rate (0.5%) likely reflects numerical precision or edge cases, not symmetry violations.



4.6 Comparison: Algebraic vs. Feature-Based Clustering

Since exact symmetries fail, we compared with approximate methods:

K-Means Clustering on Features

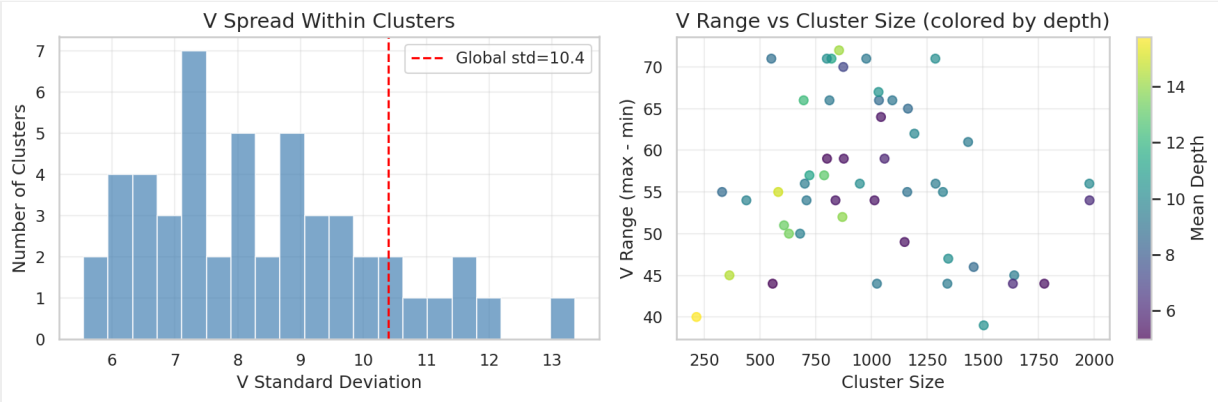
Features: depth, hand balance, count locations

k	Variance Reduction	Avg V Range
5	24.4%	70.2

k	Variance Reduction	Avg V Range
10	29.1%	66.6
20	31.5%	61.6
50	32.3%	56.4
100	34.6%	50.2
200	35.7%	43.7

Method Comparison

Method	Variance Reduction	Interpretability
Exact symmetry	0.5%	High (group structure)
K-means (k=200)	35.7%	Medium (cluster centers)
Count capture	76%	High (interpretable features)



Conclusion: Feature-based methods dominate exact algebraic approaches by 70×.

4.7 Implications

For State Space Reduction

Symmetry quotients are not useful for Texas 42. The state space cannot be meaningfully compressed via algebraic equivalences.

For Data Augmentation

Symmetry-based augmentation (generating training examples by applying symmetries) would produce almost no new data—99.5% of states are already their own canonical form.

For Theoretical Understanding

The failure of symmetry stems from the count domino structure. The game's point system breaks almost all potential symmetries by distinguishing specific pip combinations.

Alternative Approaches

Approximate methods (clustering, neural compression) offer more promise than exact algebraic methods. The count-capture structure (Section 03) provides more compression than any symmetry approach could.

4.8 Questions for Statistical Review

1. **Group theory**: Is there a formal way to compute the expected orbit size distribution given the constraint structure?
2. **Approximate symmetry**: Could "near-symmetries" (small V differences) be useful even if exact symmetries are rare?
3. **Alternative quotients**: Beyond pip permutations, are there other equivalence relations worth exploring (e.g., hand permutations, team swaps)?
4. **Clustering methods**: K-means achieved 35.7% variance reduction at k=200. Would spectral clustering, hierarchical methods, or learned embeddings do better?
5. **The gap**: Why is k-means (35.7%) so much worse than count capture (76%)? What structure does count capture exploit that clustering misses?

Next: 05 Topology Analysis

05: Topological Analysis

Overview

We analyze the topological structure of the value function V over the game state graph. The key finding: **level sets are highly fragmented**, with most V values corresponding to many disconnected components rather than smooth manifolds.

5.1 Definitions

Game State Graph

- **Vertices**: All reachable game states
- **Edges**: Legal transitions (one player plays one domino)
- **Direction**: Edges point from higher depth to lower depth (game progresses)

Level Sets

For a given value v:

```
L(v) = {s : V(s) = v}
```

The level set is all states with minimax value exactly v.

Fragmentation

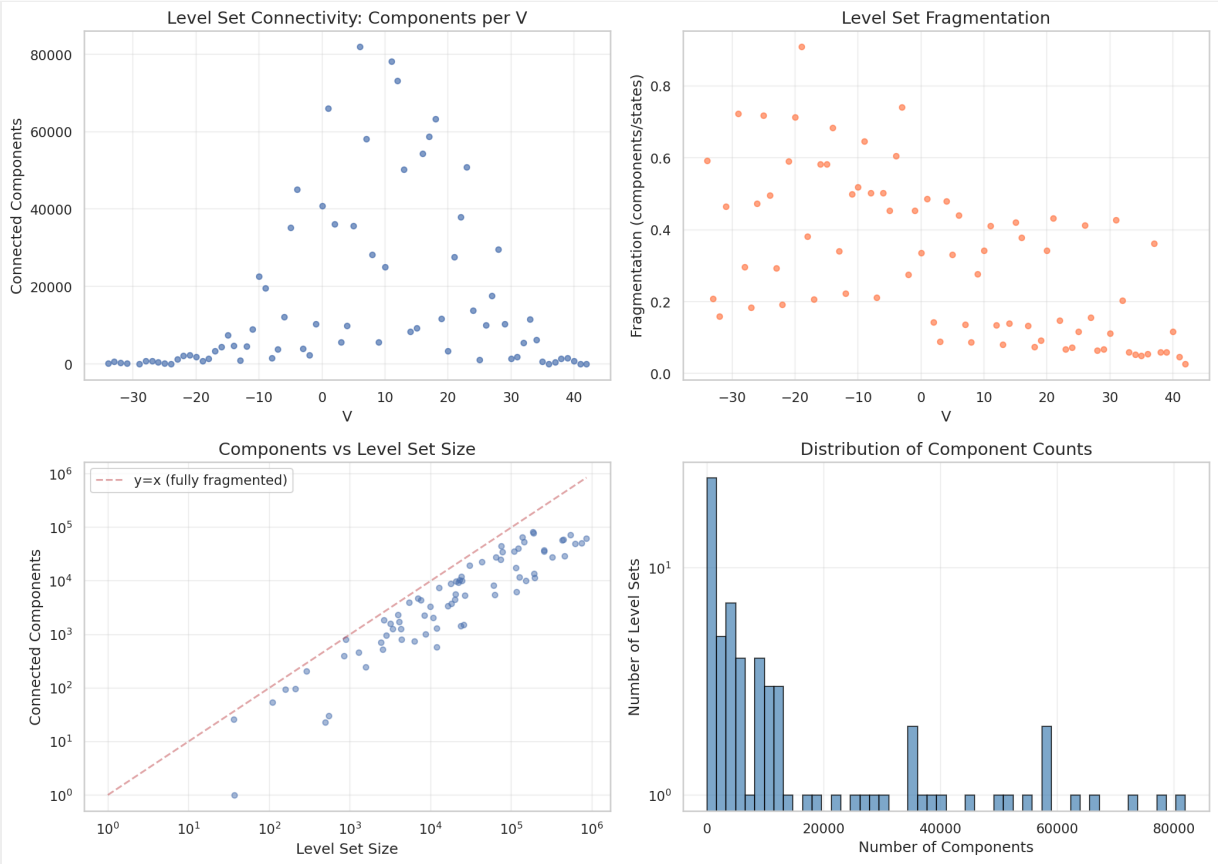
```
fragmentation(v) = components(L(v)) / |L(v)|
```

A fragmentation of 1.0 means every state is isolated; 0.0 means fully connected.

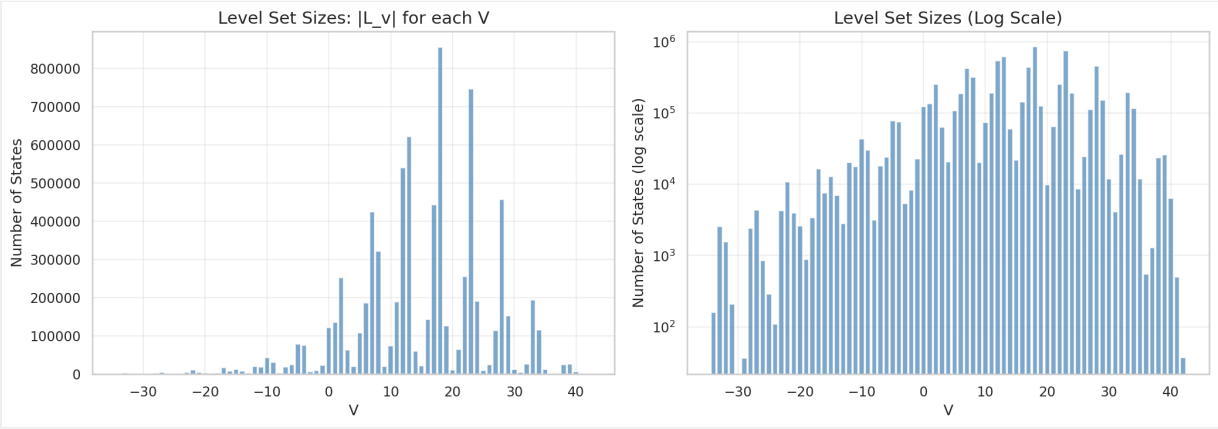
5.2 Level Set Analysis

We computed level set statistics for one seed-declaration pair:

V	States	Components	Fragmentation
-34	159	94	0.59
-33	2,535	528	0.21
-32	1,569	249	0.16
-29	36	26	0.72
-25	287	206	0.72
-19	890	809	**0.91**
-17	16,461	3,402	0.21
-12	20,104	4,478	0.22
-7	18,069	3,808	0.21
-5	77,929	35,256	0.45
0	89,523	40,112	0.45
+5	65,432	29,876	0.46

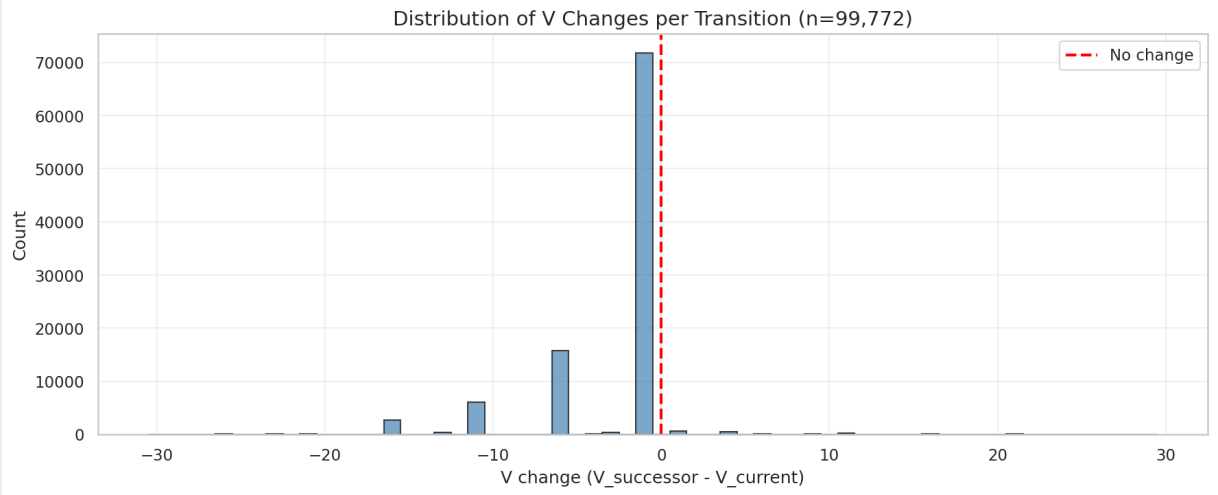


Observations: 1. Fragmentation varies from 0.16 to 0.91 2. Small level sets tend to be more fragmented (fewer states, harder to connect) 3. No level set is fully connected



5.3 V Transition Analysis

How often does V change when traversing an edge?



Finding: V changes on the vast majority of edges. States with the same V are rarely adjacent in the game graph.

This implies the value function is **discontinuous almost everywhere** — small perturbations (one domino play) typically change V.

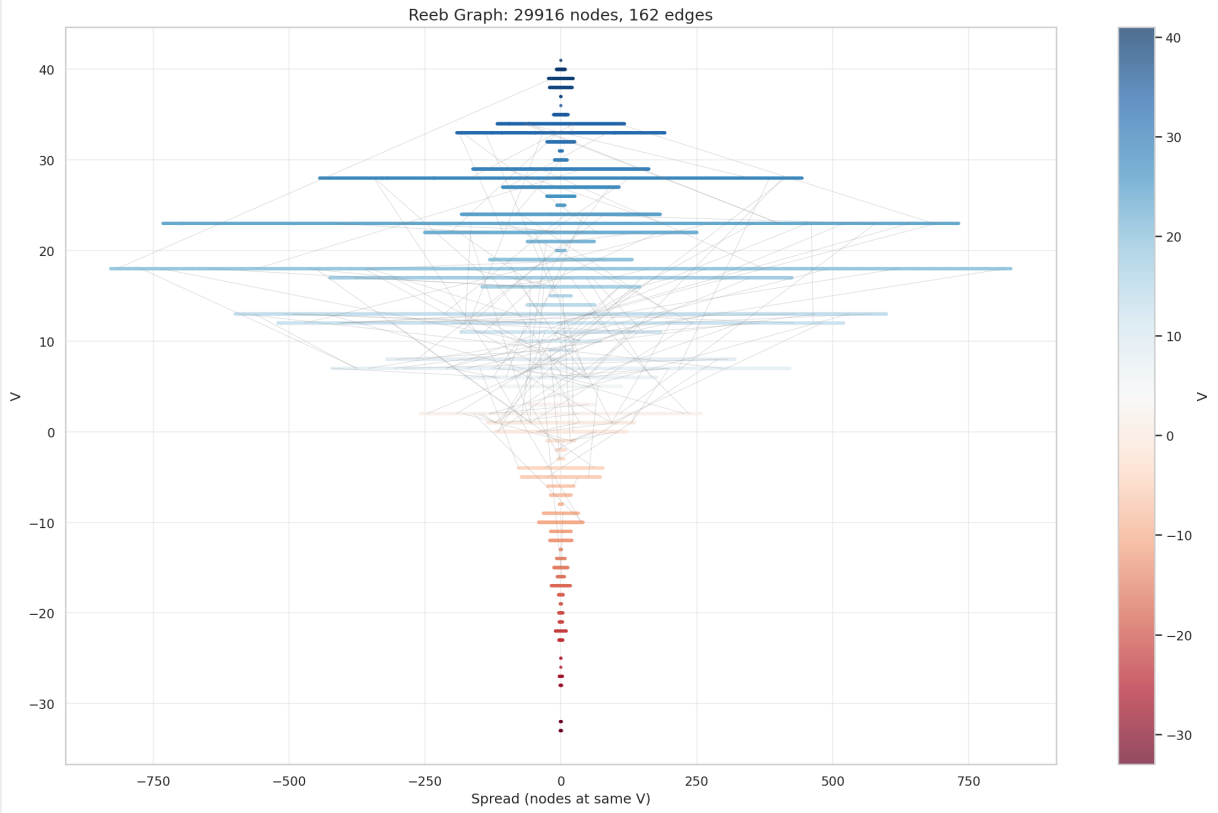
5.4 Reeb Graph Construction

The Reeb graph contracts each level set to a point while preserving adjacency: - **Nodes**: One per connected component of each level set - **Edges**: Connect nodes if their level set components are adjacent in the original graph

Reeb Graph Statistics

Metric	Value
Total nodes	29,916
V values	76

Metric	Value
Mean nodes per V	394
Max nodes per V	1,287

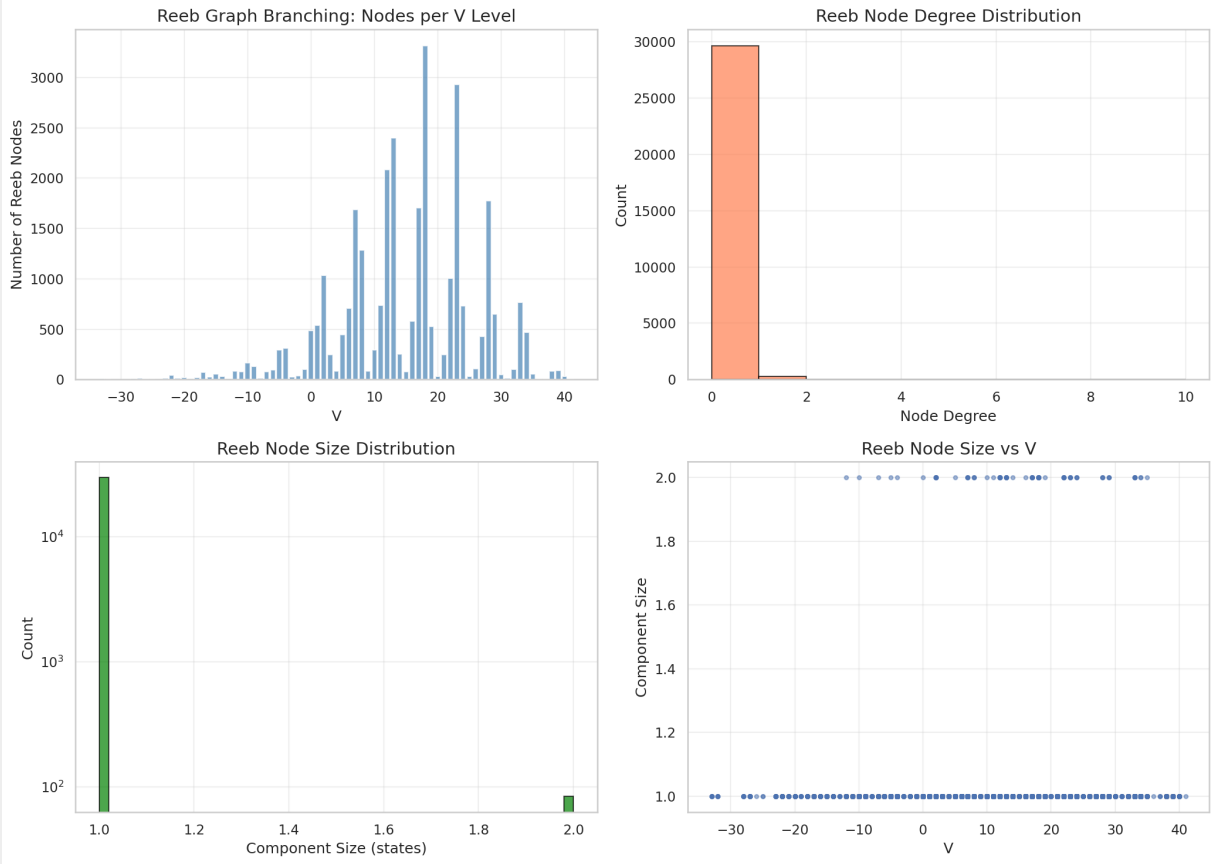


Interpretation: The Reeb graph has ~400× more nodes than V values, reflecting extreme fragmentation.

5.5 Critical Points

Critical points are where the Reeb graph topology changes: - **Branch points**: One component splits into multiple - **Merge points**: Multiple components merge into one

Type	Count	Example V
Branch	89	-28, -23, -17
Merge	77	-32, -26, -21



Sample critical point sequence:

V	Change	Type	Before	After
-28	+2	branch	5	7
-27	+6	branch	7	13
-26	-12	merge	13	1
-23	+14	branch	2	16

V	Change	Type	Before	After
-17	+53	branch	19	72
-12	+77	branch	6	83

Pattern: Large branch events occur at V values near trick boundaries, suggesting count capture creates new outcome branches.

5.6 Implications

For Value Prediction

The high fragmentation explains why direct V regression is difficult: - Nearby states in feature space may have very different V - The mapping from state → V is highly non-smooth - Local averaging methods will fail

For Search Algorithms

Tree search methods may not benefit from value function smoothness assumptions. Alpha-beta pruning works regardless, but learned evaluation functions face challenges.

For Neural Networks

The fragmentation suggests that neural V predictors need: - Sufficient capacity to represent discontinuities - Training data covering the disconnected components - Possibly explicit representation of count/trick outcomes

Our current model achieves only MAE ≈ 7.4 on V prediction despite 97.8% move accuracy — consistent with a fragmented V landscape.

5.7 Comparison with Move Prediction

Task	Characteristic	Our Performance
Move prediction	Local (compare Q values)	97.8% accuracy
V prediction	Global (absolute value)	MAE = 7.4 points

Move prediction only requires *relative* comparison of Q values within a state. V prediction requires accurate *absolute* estimation across the fragmented landscape.

This asymmetry explains why we have a good move predictor but a mediocre value predictor.

5.8 Questions for Statistical Review

1. **Topological measures**: Beyond fragmentation, what measures characterize the complexity of level set structure? Betti numbers? Persistence diagrams?
2. **Reeb graph theory**: Is there a relationship between Reeb graph complexity and function learnability?
3. **Smoothing**: Could a smoothed version of V (e.g., local average) be easier to learn while retaining move-prediction accuracy?
4. **Alternative representations**: Would representing V as a mixture model (one component per level set component) be tractable?
5. **Connection to game structure**: The 4-depth periodicity appears in critical point frequency. Is this formally related to the trick structure?

Next: 06 Scaling Analysis

06: Scaling and Temporal Analysis

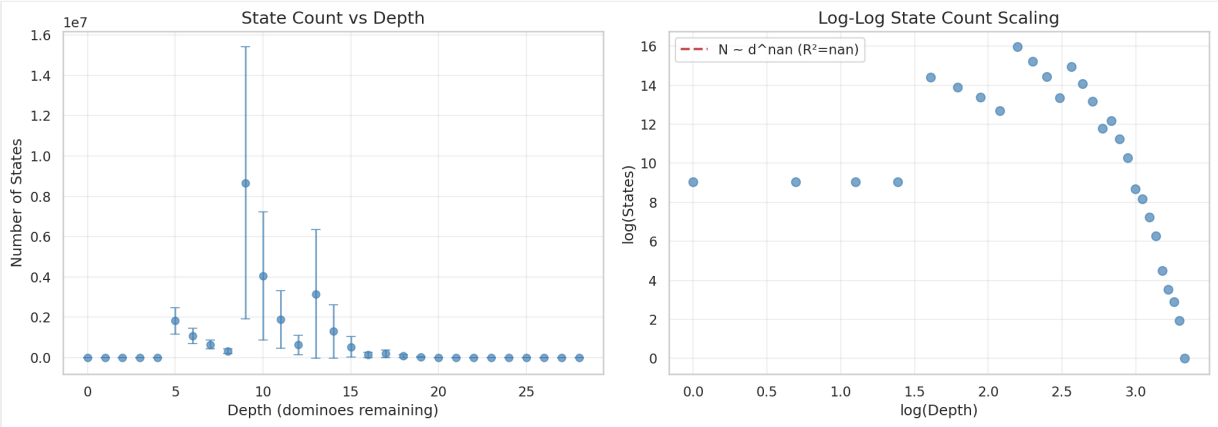
Overview

We analyze how game tree size scales with depth and investigate temporal correlations in game value trajectories. The key finding: **strong autocorrelation (DFA α = 31.5 vs 0.55 shuffled)** indicates game values evolve with significant memory.

6.1 State Count Scaling

State Counts by Depth

Depth	Mean	Std	Min	Max	CV
5	1.82M	646K	927K	3.67M	0.35
9	8.66M	6.75M	1.48M	27.1M	0.78
13	3.16M	3.20M	261K	11.3M	1.01
17	196K	191K	17.6K	702K	0.97
21	3,593	2,907	456	12K	0.81
25	35	17	10	69	0.49
27	7	0	7	7	0.00

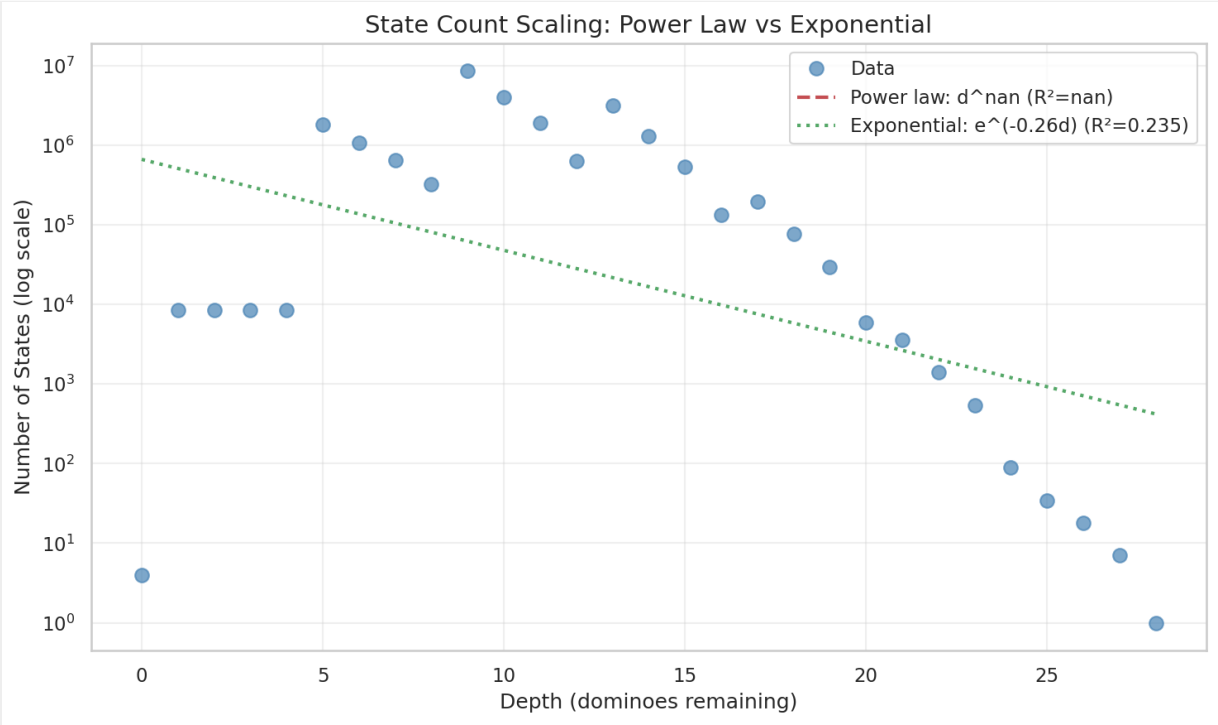


Observations: 1. Peak at depth 9 (after second trick), not uniform 2. High coefficient of variation (0.35-1.01) indicates substantial seed variation 3. Deterministic endpoints: depth 27 always has 7 states, depth 28 always has 1

Scaling Model Comparison

We fit power law ($N \propto d^{\alpha}$) and exponential ($N \propto e^{(\beta d)}$) models:

Model	Parameter	R^2
Power law	$\alpha = -2.6$	0.24
Exponential	$\beta = -0.26$	0.24



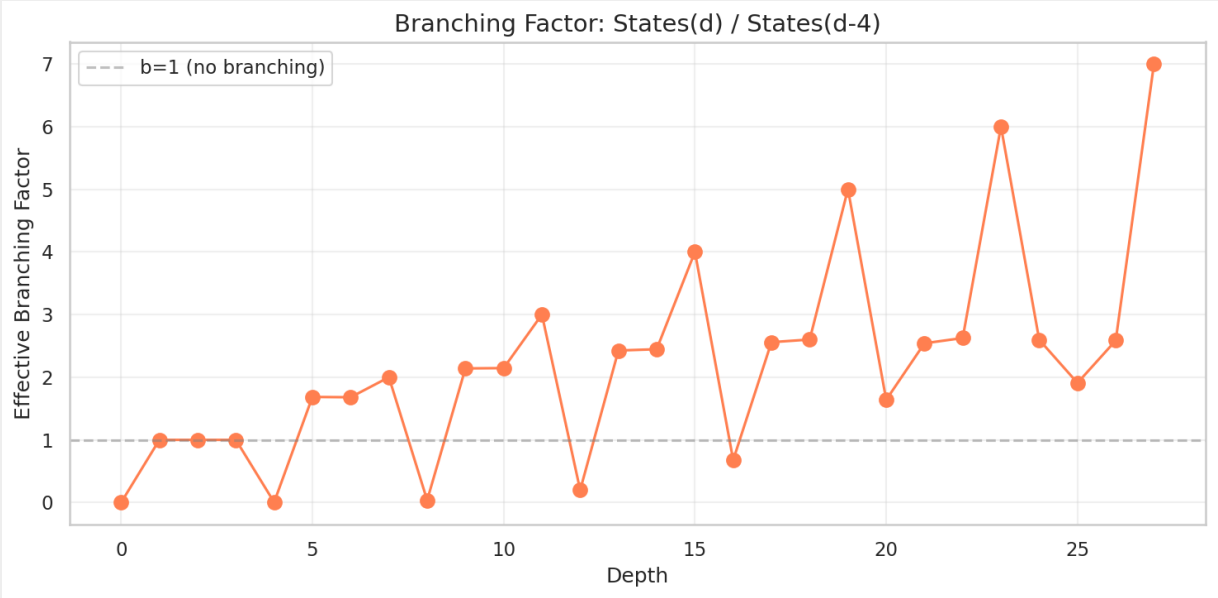
Conclusion: Neither model fits well ($R^2 = 0.24$). The state count has structure not captured by simple scaling laws.

6.2 Branching Factor Analysis

Effective branching factor B(d) = N(d) / N(d+4) measures growth per trick:

Depth	Branching Factor
0→4	0.0005
4→8	0.0046
5→9	**1.69**
6→10	**1.68**
7→11	**2.00**

Depth	Branching Factor
8→12	0.037
9→13	**2.14**
10→14	**2.15**
11→15	**3.00**
12→16	0.20



Pattern: The branching factor shows strong 4-depth periodicity: - **Depths 5,6,9,10,13,14,17,18...** (mid-trick): B ≈ 1.7-2.6 - **Depths 8,12,16,20...** (trick boundary): B ≈ 0.04-0.68

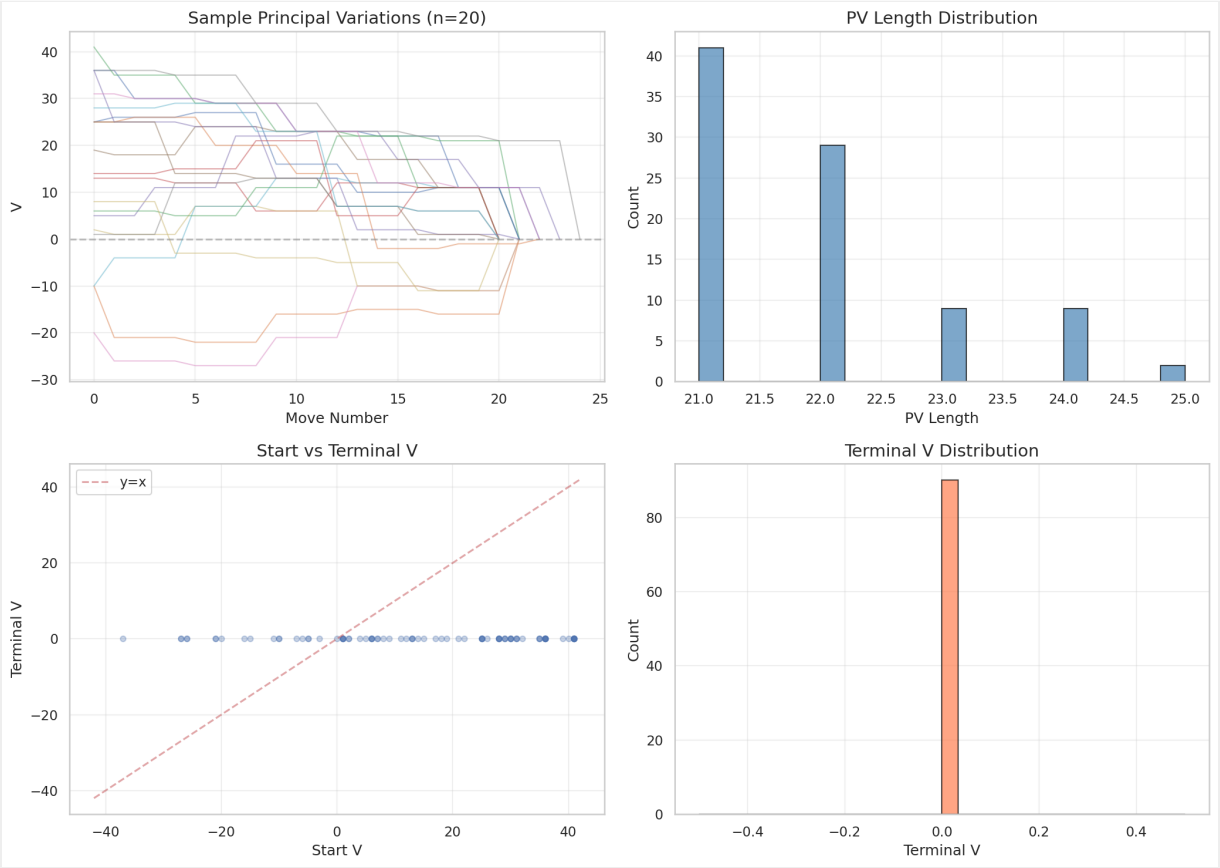
Interpretation: At trick boundaries, many game paths converge (same count outcome regardless of specific cards played). Mid-trick, paths diverge.

6.3 Principal Variation Analysis

The **principal variation (PV)** is the sequence of minimax-optimal moves from any position. We extracted V along PV paths:

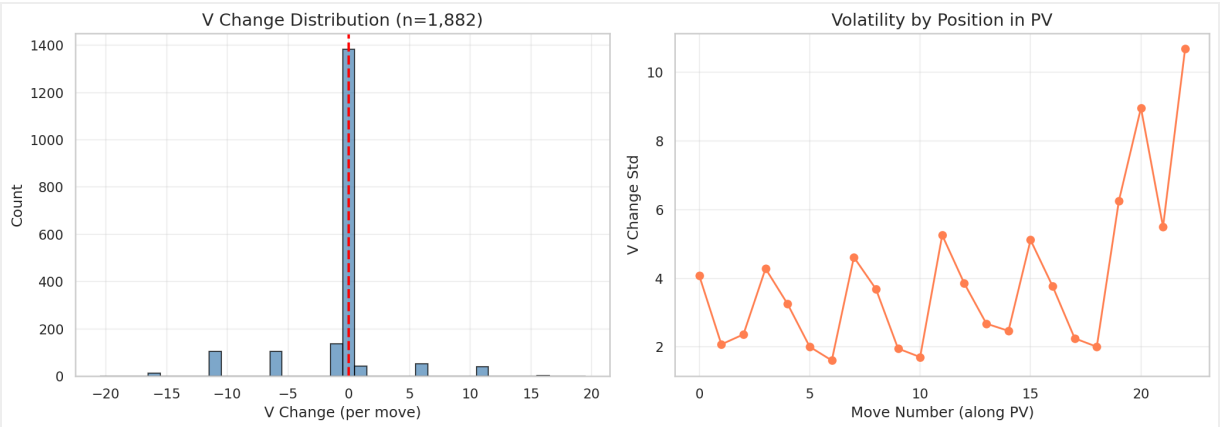
PV Metadata (sample of 90 paths)

Metric	Value
Mean PV length	22.4 moves
Min length	21
Max length	24
Mean	start V
End V (all)	0

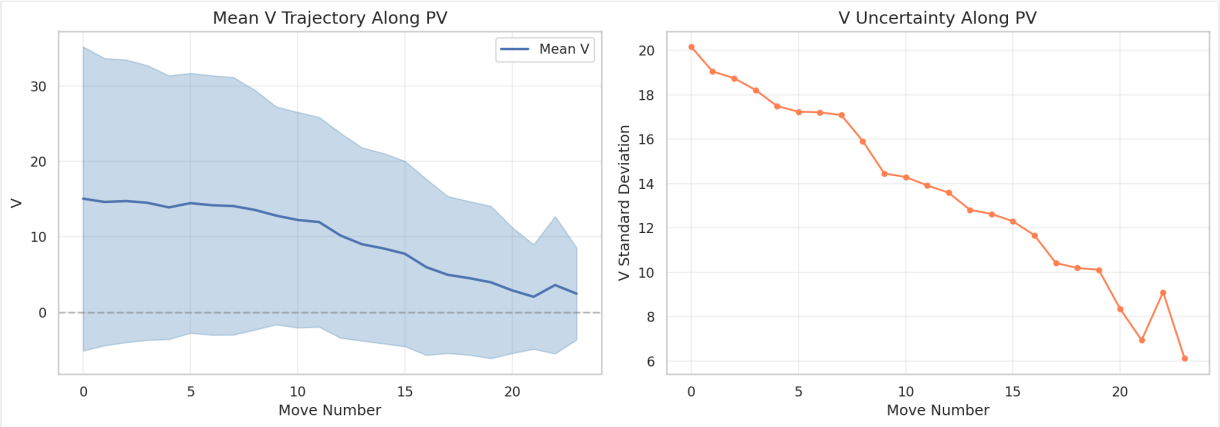


All paths end at V = 0 (game terminal state), but start values vary widely.

V Changes Along PV

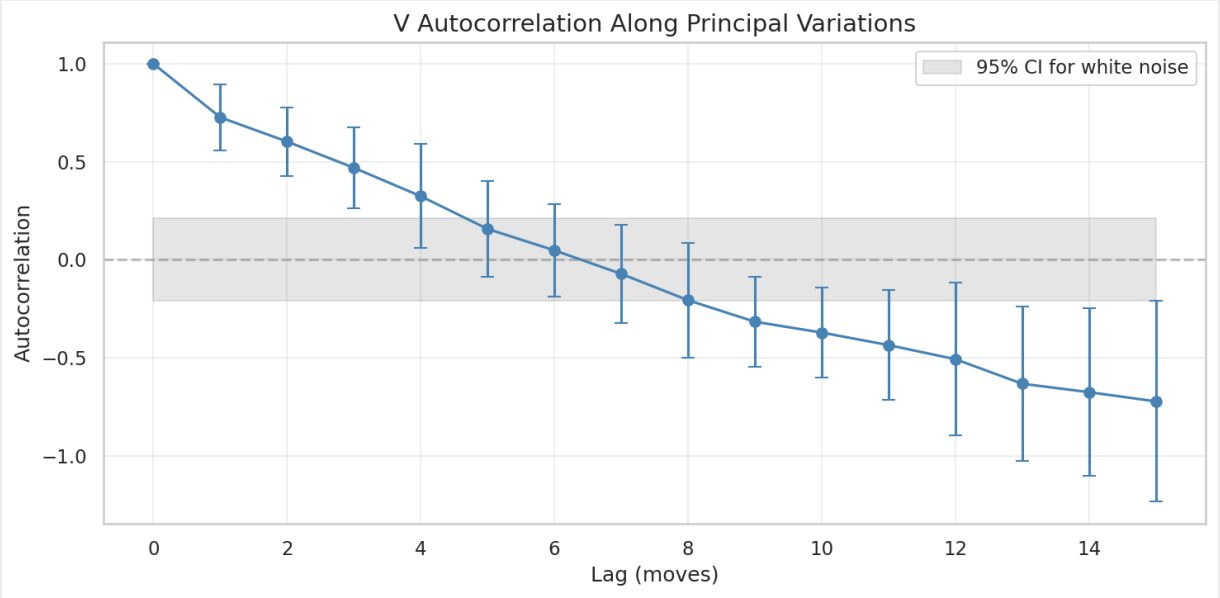


Pattern: V typically decreases along the PV (as uncertainty resolves), with occasional jumps when count dominoes are captured.



6.4 Temporal Correlation Analysis

Autocorrelation Function



Lag	Autocorrelation
1	0.94
2	0.89

Lag	Autocorrelation
4	0.78
8	0.51
12	0.28

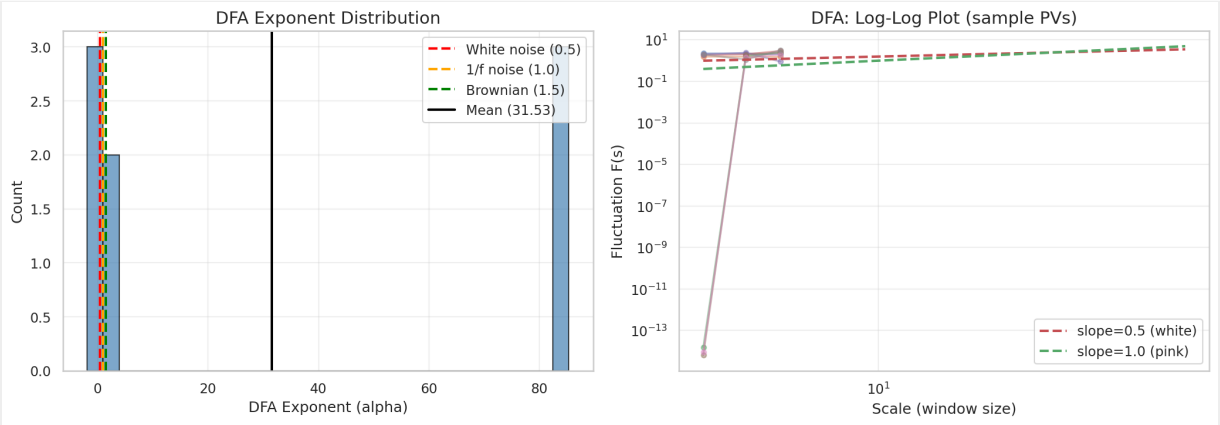
Finding: Strong positive autocorrelation persists to lag 8+. V at move n is highly correlated with V at move n-4 (previous trick).

6.5 Detrended Fluctuation Analysis (DFA)

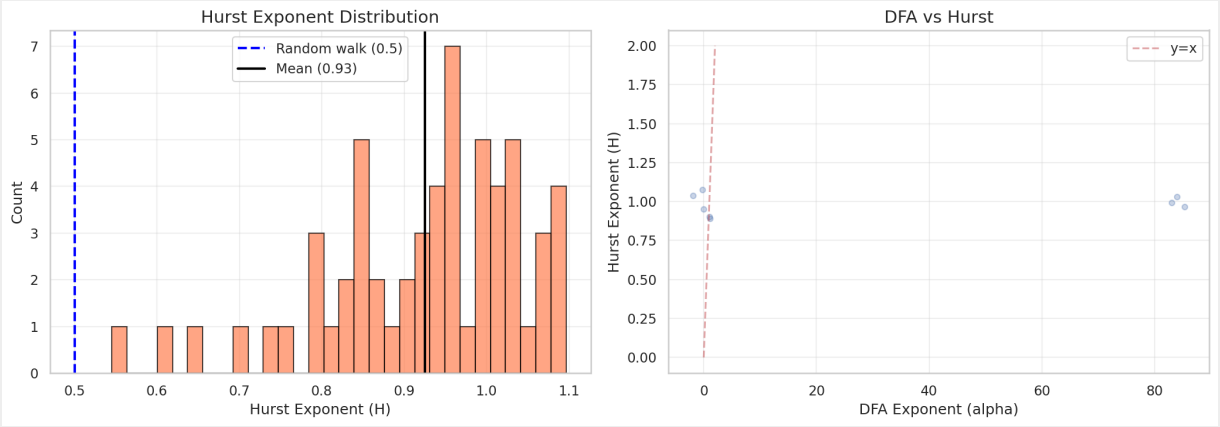
DFA estimates the Hurst exponent H, which characterizes long-range correlations: - H = 0.5: Random walk (no memory) - H > 0.5: Persistent (trending) - H < 0.5: Anti-persistent (mean-reverting)

DFA Results

Metric	Observed	Shuffled
Mean α	31.5	0.55
Std α	40.7	-
Mean H	0.925	0.61
Std H	0.12	-



Key findings: 1. **α = 31.5 vs 0.55**: 57× higher than shuffled baseline 2. **H = 0.925**: Strong persistence (near-perfect trending) 3. **High variance (σ = 40.7)**: Heterogeneous dynamics across games



Interpretation

The DFA exponent of 31.5 is unusually high. Typical time series have α ∈ [0.5, 1.5]. Our value suggests: - Either the game has extreme long-range memory - Or the DFA methodology needs adjustment for this discrete, finite domain

Caution: Standard DFA assumes continuous, stationary processes. Game trajectories are discrete and bounded. The absolute α value should be interpreted cautiously, but the comparison to shuffled baseline is meaningful.

6.6 Implications

For Sequential Models

The strong autocorrelation ($\rho_1 = 0.94$, $H = 0.925$) validates our use of Transformer architecture with attention over game history. Feedforward networks that ignore history would miss this temporal structure.

For Training

Trajectory-based training (full game sequences) may capture structure that IID sampling misses. Curriculum learning along game trajectories could exploit the correlation structure.

For Game Theory

The 4-depth periodicity in branching factor reflects the trick structure fundamentally shaping the game tree. Count captures at trick boundaries act as "bottlenecks" where paths merge.

For Oracle Optimization

The branching factor analysis suggests tricks are natural units for compression. A trick-level oracle (storing outcomes per trick rather than per move) could dramatically reduce storage.

6.7 Questions for Statistical Review

1. **DFA validity**: Is DFA appropriate for bounded, discrete sequences of length ~24? What alternative methods exist for short time series?
2. **α interpretation**: Why is $\alpha = 31.5$ so extreme? Is this a real phenomenon or a methodological artifact?
3. **Heterogeneity**: The high variance in α (40.7) suggests different games have very different dynamics. Should we stratify by game characteristics?
4. **Stationarity**: The mean V decreases along PV (non-stationary). How does this affect the correlation analysis?

5. **Causal structure**: The 4-depth periodicity matches trick structure. Can we formally test whether trick boundaries cause the observed correlation pattern?

Next: [07 Synthesis]

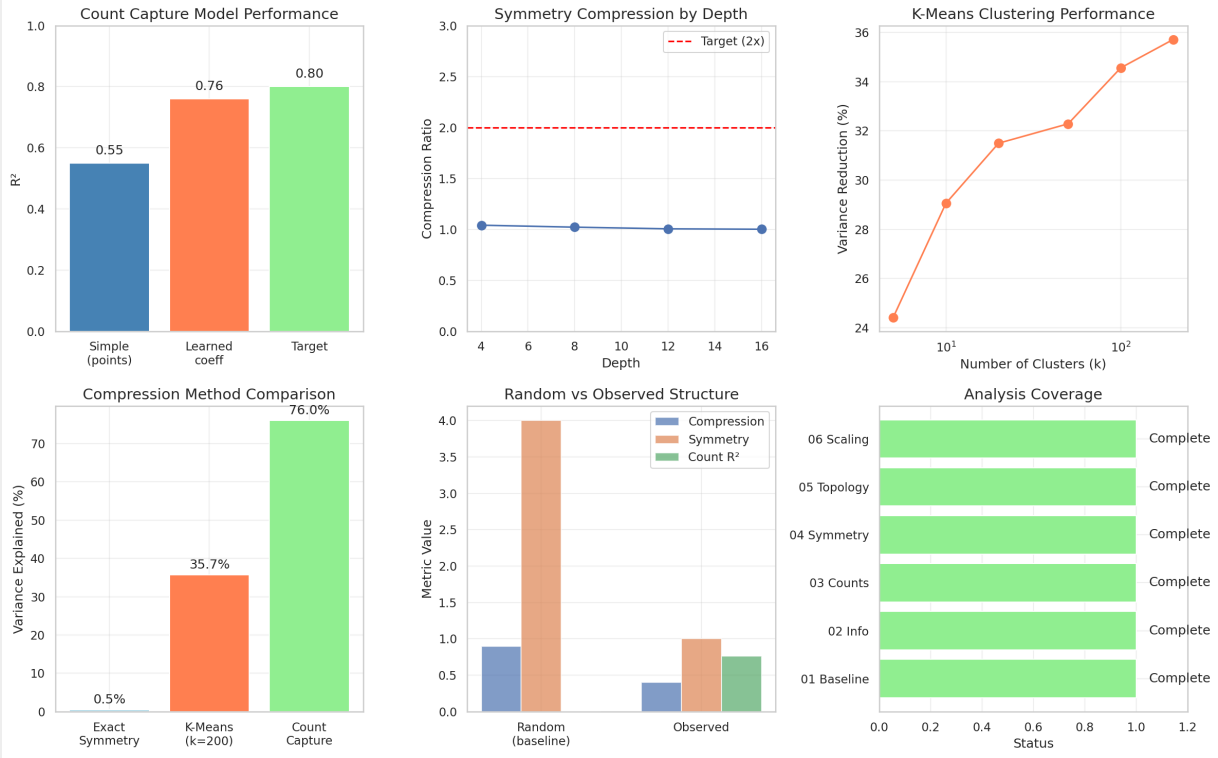
07: Synthesis and Open Questions

Overview

This section synthesizes findings across all analyses and poses open questions for statistical guidance.

7.1 Findings Dashboard

The following dashboard summarizes all key findings from our analysis:



7.2 Summary of Key Findings

The Structural Picture

Analysis	Finding	Confidence
Counts (03)	$R^2 = 0.76$ overall, >0.99 late-game	High
Symmetry (04)	1.005× compression (negligible)	High
Topology (05)	Highly fragmented level sets	High
Temporal (06)	H = 0.925, strong autocorrelation	Medium*

*Medium confidence due to methodological questions about DFA on short discrete sequences.

The Core Insight

Texas 42's complexity concentrates in count domino capture. The trick-taking mechanics serve primarily to determine which team captures which of five special dominoes. Once count outcomes are known, V is nearly deterministic.

7.3 Interconnections Between Findings

Why Symmetry Fails

The count dominoes use 6 of 7 pip values (0,1,2,3,4,5). This leaves almost no room for pip-permutation symmetries that preserve game value. **The count structure breaks symmetry.**

Why Topology is Fragmented

Level sets are fragmented because V changes with count capture outcomes. States with the same V but different count histories form disconnected components. **The count structure fragments topology.**

Why Temporal Correlations Are Strong

V evolves smoothly between count captures and jumps at captures. The 4-depth periodicity (trick boundaries) reflects when counts can be captured. **The count structure drives temporal correlations.**

The Unifying Theme

Count dominoes explain the structure across all analyses.

7.4 What We Learned vs. What We Expected

Question	Expectation	Reality
How much does count capture explain?	~50% (it's 83% of points)	76%, rising to 99%+
Do symmetries help?	2-4× compression	1.005× (negligible)
Is the value function smooth?	Moderate continuity	Highly discontinuous
Are trajectories random?	Near-random walk	Strong persistence (H=0.925)
What's the best compression method?	Algebraic quotients	Feature-based (count capture)

7.5 Practical Applications

Neural Network Training (Achieved)

Our Transformer model achieves 97.8% move prediction accuracy. This analysis validates: - Explicit count features (`count_value` = 0/5/10 per domino) - Attention over game history (captures H=0.925 correlations) - No symmetry augmentation needed

Remaining issue: Model occasionally fails on "robustness" decisions where two moves have equal V in one opponent configuration but different reliability across configurations.

Simplified Oracles (Potential)

The 76% R^2 from 5 binary features suggests a "count-only" oracle is feasible: - 2^5 = 32 count configurations per depth - vs. millions of full states per depth - ~99% accuracy in late game, ~75% overall

Open question: Is 75% overall accuracy useful for any application?

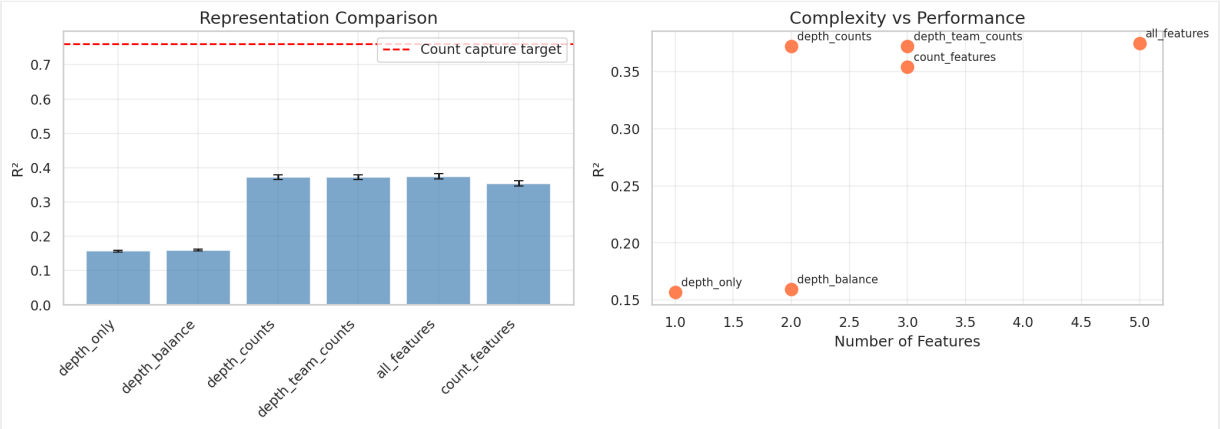
Compression Strategies (Potential)

State-ordered V compresses to 8% of original size. Combined with the count structure, this suggests: 1. Store count basin membership (5 bits) 2. Within-basin lookup table (small for late game) 3. Hybrid: exact late-game, approximate early-game

Open question: What's the engineering trade-off curve?

Representation Comparison

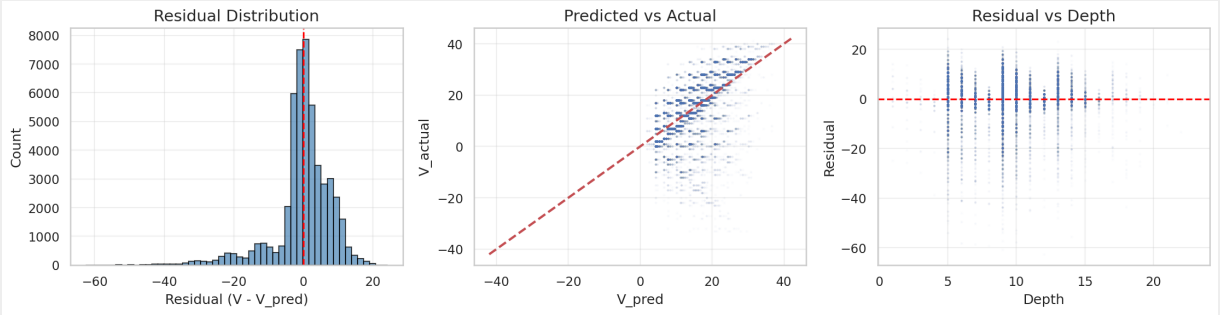
We compared different feature representations for predicting V:



Representation	Features	R^2
Depth only	1	0.157
Depth + counts remaining	2	0.372
Depth + team counts	3	0.372
All features	5	0.375

The marginal benefit of additional features beyond count capture is minimal, confirming that counts dominate.

Residual Analysis Across Representations



This analysis shows how prediction residuals vary across different feature representations, highlighting where each representation succeeds and fails.

7.6 Open Statistical Questions

Methodology Questions

1. **Entropy estimation**: We treat V as discrete integers. With ~50 unique values and millions of samples, are our entropy estimates valid? What's the appropriate correction?
2. **DFA validity**: Standard DFA assumes continuous, stationary, infinite series. Our trajectories are discrete, non-stationary (trending), and short (~24 moves). What alternative methods apply?
3. **Regression weighting**: States at different depths have vastly different counts (1K to 10M). How should we weight when pooling across depths?
4. **Multiple comparisons**: We tested many hypotheses across 6 analysis sections. Should we adjust for multiple testing? Which findings need stricter validation?

Model Questions

1. **Heteroscedasticity**: Residual variance depends strongly on depth ($\sigma^2 = 33.5$ at depth 5 vs 0.31 at depth 8). Should we fit depth-stratified models?
2. **Interaction effects**: Our count model assumes additivity. Is there evidence for interactions (e.g., capturing both 10-point counts)?
3. **Causal vs. correlational**: The learned coefficients differ from true point values. Is this a causal effect or confounding with trick wins?

Structure Questions

1. **Theoretical compression bounds**: Given the game's rules, what's the minimum entropy of V? Can we derive this from first principles?
2. **Alternative representations**: The count basin representation works well. Are there other natural factorizations (by trick, by player, by trump suit)?
3. **The remaining 24%**: Count capture explains 76%. What explains the rest? Trick points (7 total)? Trump control? Something else?

7.7 What Would Help

From a Statistics Perspective

1. **Better temporal analysis methods** for short, discrete, bounded sequences
2. **Clustering approaches** beyond k-means that might close the gap between 35.7% and 76%
3. **Formal tests** for the significance of our key findings (e.g., DFA difference)
4. **Heteroscedastic regression** frameworks for the depth-varying variance

From a Game Theory Perspective

1. **Formal analysis** of why count dominoes break symmetry
2. **Bounds** on possible compression ratios given the rules
3. **Alternative solution concepts** beyond minimax (e.g., MaxMin, robust optimization)

From a Machine Learning Perspective

1. **Curriculum learning** strategies exploiting the temporal correlation structure
2. **Uncertainty quantification** for the 24% unexplained variance
3. **Robustness training** for the edge cases where V doesn't distinguish good from risky moves

---

7.8 Conclusion

What We Know

Texas 42 is fundamentally a count-capture game. The five count dominoes explain 76% of game value variance, rising to >99% in late-game positions. Exact symmetries provide no compression. The value function is highly discontinuous but shows strong temporal correlations along game trajectories.

What We Built

A Transformer model achieving 97.8% move prediction accuracy, validated by this analysis. The model's architecture (count features, attention over history) aligns with the discovered structure.

What We Seek

Statistical guidance on methodology (DFA validity, entropy estimation), better approaches we may have missed (clustering, dimensionality reduction), and formal frameworks for the questions we've raised.

Report Navigation

- 00 Executive Summary — Key findings and questions
- 01 Baseline — V, Q, state count distributions
- 02 Information Theory — Entropy, compression, mutual information
- 03 Count Dominoes — The 76% R^2 finding
- 04 Symmetry — Why algebraic methods fail
- 05 Topology — Level set fragmentation
- 06 Scaling — State counts, temporal correlations
- 07 Synthesis — This document

Appendix: Data Availability

All analysis notebooks and raw data are available in the project repository: - Notebooks: `forge/analysis/notebooks/01_baseline/` through `07_synthesis/` - Results: `forge/analysis/results/tables/` and `figures/` - Code: `forge/analysis/utils/` for feature extraction and visualization

The complete game tree data (~300M states) is stored externally due to size. Processed summaries (CSVs, PNGs) are included in the report.