Title

A Structural Framework for Nonlinear Structural Systems

Abstract

This paper presents a computational framework for analyzing structured systems using the Voynich Manuscript as a case study.
We identify consistent morphological, positional, and topological constraints that cannot be explained by known linguistic or cryptographic models.
Using reproducible analytical methods, we demonstrate stable internal structure across domains.
These findings suggest the manuscript exhibits structured constraint consistent with a non-linear structural system rather than a cipher or natural language.
The framework is designed to be testable, extensible, and falsifiable.

1. Problem Statement

For centuries, the Voynich Manuscript has defied all conventional linguistic and cryptographic methods of interpretation. Traditional approaches assume it encodes either a natural language or a substitution cipher. However, these approaches fail because they do not account for the manuscript's unusual statistical consistency, context-dependent glyph patterns, and diagram-linked relational structures.

This paper proposes a fundamentally different perspective: that Voynichese is a formal symbol system exhibiting structured relational constraints whose structure is distributed across morphological, positional, and topological features rather than linear text. By recognizing the manuscript as a multi-layer nonlinear structural system, we can model its internal coherence where other methods fail.

1.1 Contextual Positioning

This work situates itself at the intersection of structural analysis, information theory, and computational modeling. Unlike prior approaches that seek linguistic decoding or cryptographic resolution, the present framework treats the Voynich Manuscript as a structured system governed by internal constraints. The analysis builds upon established observations of statistical regularity while extending them through a formal, testable model of non-linear relational structure organization. By focusing on structural behavior rather than symbolic interpretation, this work reframes the manuscript as a computational artifact whose coherence can be empirically evaluated independent of linguistic assumptions.

2. Observed Constraints

The Voynich Manuscript exhibits a set of structural properties that remain stable across pages, sections, and scribal variations. These properties impose constraints that any valid interpretive framework must satisfy.

2.1 Statistical Regularity

- Glyph distributions remain remarkably stable across the manuscript.
- Positional entropy is non-uniform, with constrained behavior at token boundaries.

- Repetition patterns are structured rather than random.

These properties are inconsistent with random text generation and are not readily explained by known cipher systems.

2.2 Positional Dependence

- Glyph behavior is strongly position-dependent within tokens.
- Prefix, medial, and suffix regions exhibit distinct statistical roles.
- This behavior persists across different manuscript sections.

Such consistency is incompatible with phonetic encoding or ad hoc symbol substitution.

2.3 Cross-Domain Stability

- The same structural patterns appear in botanical, astronomical, and diagrammatic sections.
- Visual layout and textual structure reinforce one another.
- Structural roles remain invariant even when visual content changes.

This implies a shared underlying system rather than multiple unrelated encodings.

2.4 Constraint Summary

Any valid model of the Voynich Manuscript must account for:

- Non-random statistical structure
- Positionally constrained symbol behavior
- Cross-domain structural coherence
- Stability under transformation

Models failing these criteria cannot explain the manuscript’s behavior.

3. Formal Model

3.1 Symbol Set Definition

Let

- $\Sigma=\{s1,s2,...,sn\}$ be the finite set of glyphs observed in the corpus.
- Tokens are ordered sequences $T=(s1,s2,...,sk)$, where $k\geq1$.

Tokens are treated as structured units rather than linear strings.

3.2 Token Decomposition

Each token is decomposed into three functional regions:

$T=P\cdot C\cdot S$

Where:

- P: prefix operator set

- C: core relational structure unit
- S: suffix or state modifier

Membership in each region is determined empirically through positional entropy minimization and frequency clustering.

3.3 Positional Function Mapping

Define a positional function:

$f:(s_i, pos_i) \rightarrow r_j$

Where:

- $s_i \in \Sigma$
- $pos_i \in \{1,\ldots,k\}$
- $r_j \in \{P,C,S\}$

The mapping is stable across sections and independent of visual context.

3.4 State-Dependent Behavior

Suffix elements encode state transitions applied to core units.

Let:

$C' = \delta(C,S)$

Where:

- δ is a deterministic state transition function
- C' is the modified relational structure state

This behavior aligns with finite-state systems rather than phonological grammar.

3.5 Cross-Domain Invariance

For any domain $D \in \{botanical, astronomical, balneological, diagrammatic\}$:

$\Phi_D(P,C,S) \approx \Phi_{D'}(P,C,S)$

where Φ denotes structural behavior under domain variation.

This invariance implies a single generative mechanism across domains.

3.6 System Properties

The resulting system exhibits:

- Non-linearity
- Deterministic local transitions
- Global structural stability
- Context-sensitive behavior

These properties are inconsistent with substitution ciphers or phonetic languages.

3.7 Model Classification

The system is best characterized as:

- A **behavior consistent with a non-linear state transition system**, or
- A **structural relational system with constrained transitions**

It is not:

- A natural language
- A monoalphabetic cipher
- A stochastic text generator

3.8 Testability

The model predicts:

1. Recurrent positional constraints across all sections
2. Stable transition probabilities between functional classes
3. Invariance under permutation of visual domains

Violation of these predictions falsifies the model.

> |Structural invariance and constrained entropy behavior are consistent with established principles of information theory (Shannon, 1948; Cover & Thomas, 2006).|

The present framework does not assume intentional encoding or semantic reference; it models only the structural constraints that govern symbol arrangement.

Algorithm 1 — Formalization of the Voynich Structural Relational Model

```
# ==============================================
# Voynich Framework — Formal Notation / Pseudocode
# Section 3 (Formal Model) executable skeleton
# ==============================================


from dataclasses import dataclass
from typing import Dict, List, Tuple, Set, Optional
import math


Glyph = str
Token = List[Glyph]
Corpus = List[Token]
Domain = str  # {"botanical","astronomical","balneological","diagrammatic",...}
```

```
# ----------------------------
# 3.1 Symbol set and corpus
# ----------------------------

def build_alphabet(corpus: Corpus) -> Set[Glyph]:
    Σ: Set[Glyph] = set()
    for tok in corpus:
        Σ.update(tok)
    return Σ



# ----------------------------
# 3.2 Shannon entropy by token position
# ----------------------------

def shannon_entropy(counts: Dict[Glyph, int]) -> float:
    total = sum(counts.values())
    if total <= 0:
        return 0.0
    H = 0.0
    for c in counts.values():
        p = c / total
        if p > 0:
            H -= p * math.log2(p)
    return H

def positional_counts(corpus: Corpus) -> Dict[int, Dict[Glyph, int]]:
    # counts[pos][glyph] = count
    counts: Dict[int, Dict[Glyph, int]] = {}
    for tok in corpus:
        for pos, g in enumerate(tok, start=1):
            counts.setdefault(pos, {})
```

```
            counts[pos][g] = counts[pos].get(g, 0) + 1
    return counts


def positional_entropy(corpus: Corpus) -> Dict[int, float]:
    counts = positional_counts(corpus)
    return {pos: shannon_entropy(gcounts) for pos, gcounts in
counts.items()}



# ----------------------------
# 3.3 Region assignment P/C/S
# ----------------------------


@dataclass(frozen=True)
class Regions:
    # Region boundaries are learned from entropy minima / ridges
    # Example: prefix = positions <= p_end, suffix = positions >=
s_start
    p_end: int
    s_start: int


def learn_region_boundaries(entropy_by_pos: Dict[int, float]) ->
Regions:
    # Minimal formal version:
    # - core region centered on entropy minimum
    # - prefix and suffix are the high-entropy edges around it
    #
    # Replace with: robust trough detection + width selection.
    positions = sorted(entropy_by_pos.keys())
    core_center = min(positions, key=lambda p: entropy_by_pos[p])


    # Choose simple boundaries (tunable):
    # prefix ends just before core_center, suffix starts just after
core_center
```

```
    p_end = max(1, core_center - 1)
    s_start = min(max(positions), core_center + 1)

    return Regions(p_end=p_end, s_start=s_start)

def assign_region(pos: int, k: int, regions: Regions) -> str:
    # r_j ∈ {P, C, S}
    if pos <= regions.p_end:
        return "P"
    if pos >= regions.s_start:
        return "S"
    return "C"



# ----------------------------
# 3.2/3.3 Token decomposition T = P·C·S
# ----------------------------

@dataclass
class Decomposition:
    P: Token
    C: Token
    S: Token

def decompose_token(tok: Token, regions: Regions) -> Decomposition:
    P, C, S = [], [], []
    k = len(tok)
    for pos, g in enumerate(tok, start=1):
        r = assign_region(pos, k, regions)
        if r == "P":
            P.append(g)
        elif r == "C":
            C.append(g)
```

```
        else:
            S.append(g)
    return Decomposition(P=P, C=C, S=S)



# ----------------------------
# 3.4 State-dependent behavior: C' = δ(C, S)
# ----------------------------

@dataclass(frozen=True)
class ProtoForms:
    # Abstract ids; populate via clustering / mapping
    O: Optional[str]  # operator class
    M: Optional[str]  # meaning-core class
    S: Optional[str]  # state class


def classify_operator(prefix: Token, operator_vocab: Dict[Tuple[Glyph,
...], str]) -> Optional[str]:
    # Map observed prefixes to operator classes (O₁..O₄)
    key = tuple(prefix)
    return operator_vocab.get(key)


def classify_core(core: Token, core_clusterer) -> Optional[str]:
    # Placeholder: cluster assignment
    # Implement via co-occurrence vectors / k-means, etc.
    return core_clusterer(core)


def classify_state(suffix: Token, state_vocab: Dict[Tuple[Glyph, ...],
str]) -> Optional[str]:
    key = tuple(suffix)
    return state_vocab.get(key)


def delta(core_class: str, state_class: str) -> str:
```

```
    # δ is a deterministic transition function over core-states
    # Minimal form: return a labeled transition; replace with real
state machine.
    return f"{core_class}->{state_class}"


def apply_state(core_class: Optional[str], state_class: Optional[str])
-> Optional[str]:
    if core_class is None or state_class is None:
        return None
    return delta(core_class, state_class)



# ----------------------------
# 3.5 Cross-domain invariance check
# Φ_D(P,C,S) ≈ Φ_D'(P,C,S)
# ----------------------------


@dataclass
class DomainStats:
    # Minimal measurable invariants:
    # - prefix frequency distribution
    # - core cluster distribution
    # - suffix frequency distribution
    # - transition matrix P->C->S patterns
    prefix_freq: Dict[str, int]
    core_freq: Dict[str, int]
    suffix_freq: Dict[str, int]
    transition_freq: Dict[str, int]


def compute_domain_stats(
    corpus: Corpus,
    regions: Regions,
    operator_vocab: Dict[Tuple[Glyph, ...], str],
```

```
    core_clusterer,
    state_vocab: Dict[Tuple[Glyph, ...], str],
) -> DomainStats:
    prefix_freq: Dict[str, int] = {}
    core_freq: Dict[str, int] = {}
    suffix_freq: Dict[str, int] = {}
    transition_freq: Dict[str, int] = {}

    for tok in corpus:
        dec = decompose_token(tok, regions)
        O = classify_operator(dec.P, operator_vocab) or "O_?"
        M = classify_core(dec.C, core_clusterer) or "M_?"
        S = classify_state(dec.S, state_vocab) or "S_?"

        prefix_freq[O] = prefix_freq.get(O, 0) + 1
        core_freq[M] = core_freq.get(M, 0) + 1
        suffix_freq[S] = suffix_freq.get(S, 0) + 1

        # capture compositional signature
        sig = f"{O}({M})^{S}"
        transition_freq[sig] = transition_freq.get(sig, 0) + 1

    return DomainStats(
        prefix_freq=prefix_freq,
        core_freq=core_freq,
        suffix_freq=suffix_freq,
        transition_freq=transition_freq,
    )

def distance_L1(a: Dict[str, int], b: Dict[str, int]) -> float:
    keys = set(a) | set(b)
    return sum(abs(a.get(k, 0) - b.get(k, 0)) for k in keys)
```

```
def invariance_test(stats_A: DomainStats, stats_B: DomainStats, threshold: float) -> bool:
    # Minimal invariance criterion using L1 distances (replace with KL/JS divergence)
    dP = distance_L1(stats_A.prefix_freq, stats_B.prefix_freq)
    dC = distance_L1(stats_A.core_freq, stats_B.core_freq)
    dS = distance_L1(stats_A.suffix_freq, stats_B.suffix_freq)
    dT = distance_L1(stats_A.transition_freq, stats_B.transition_freq)
    return (dP + dC + dS + dT) <= threshold


# ----------------------------
# 3.6/3.7 Model classification & falsifiability checks
# ----------------------------

@dataclass
class Predictions:
    # Empirical predictions derived from the model
    positional_entropy_profile: Dict[int, float]
    stable_region_boundaries: Regions
    stable_operator_inventory: Set[str]
    stable_state_inventory: Set[str]

def build_predictions(
    corpus: Corpus,
    operator_vocab: Dict[Tuple[Glyph, ...], str],
    core_clusterer,
    state_vocab: Dict[Tuple[Glyph, ...], str],
) -> Predictions:
    Hpos = positional_entropy(corpus)
    regions = learn_region_boundaries(Hpos)

    # infer inventories from corpus
```

```
    ops: Set[str] = set()
    sts: Set[str] = set()
    for tok in corpus:
        dec = decompose_token(tok, regions)
        ops.add(classify_operator(dec.P, operator_vocab) or "O_?")
        sts.add(classify_state(dec.S, state_vocab) or "S_?")

    return Predictions(
        positional_entropy_profile=Hpos,
        stable_region_boundaries=regions,
        stable_operator_inventory=ops,
        stable_state_inventory=sts,
    )

def falsify_if_violated(pred: Predictions, observed: Predictions,
tol_entropy: float) -> List[str]:
    failures: List[str] = []

    # 1) entropy profile stability
    keys = set(pred.positional_entropy_profile) &
set(observed.positional_entropy_profile)
    for pos in keys:
        if abs(pred.positional_entropy_profile[pos] -
observed.positional_entropy_profile[pos]) > tol_entropy:
            failures.append(f"Entropy mismatch at position {pos}")

    # 2) region boundary stability
    if pred.stable_region_boundaries !=
observed.stable_region_boundaries:
        failures.append("Region boundary mismatch (P/C/S segmentation
unstable)")

    # 3) inventory stability
```

```
    if not
pred.stable_operator_inventory.issubset(observed.stable_operator_inven
tory | pred.stable_operator_inventory):
        failures.append("Operator inventory instability")
    if not
pred.stable_state_inventory.issubset(observed.stable_state_inventory |
pred.stable_state_inventory):
        failures.append("State inventory instability")

    return failures


# ----------------------------
# Minimal "run" scaffold
# ----------------------------

def pipeline(corpus_by_domain: Dict[Domain, Corpus], operator_vocab,
core_clusterer, state_vocab):
    # Build baseline predictions on full corpus or a designated
reference domain
    full_corpus = [tok for dom in corpus_by_domain for tok in
corpus_by_domain[dom]]
    baseline = build_predictions(full_corpus, operator_vocab,
core_clusterer, state_vocab)
    regions = baseline.stable_region_boundaries

    # Domain statistics
    domain_stats: Dict[Domain, DomainStats] = {}
    for dom, corp in corpus_by_domain.items():
        domain_stats[dom] = compute_domain_stats(corp, regions,
operator_vocab, core_clusterer, state_vocab)

    # Cross-domain invariance checks (pairwise)
    domains = list(corpus_by_domain.keys())
    invariance_matrix: Dict[Tuple[Domain, Domain], bool] = {}
    for i in range(len(domains)):
```

```
        for j in range(i + 1, len(domains)):
            A, B = domains[i], domains[j]
            invariance_matrix[(A, B)] =
invariance_test(domain_stats[A], domain_stats[B], threshold=1_000.0)


    return baseline, domain_stats, invariance_matrix
```

This section operationalizes the framework as a reproducible pipeline: (i) build a corpus of EVA tokens, (ii) compute positional statistics over token indices, (iii) apply a fixed P–C–S segmentation, and (iv) test whether derived structural signatures persist under domain stratification. The figures in Section 4 are positional (not referenced in-text) but are generated directly from the accompanying scripts using shared parameters, enabling independent replication.

Importantly, the analysis targets structural behavior, constraints, transitions, and invariants, rather than semantic translation. The framework makes falsifiable claims about measurable regularities (entropy profiles, conserved domain signatures, and directional transition structure). If those regularities fail to reproduce on an independent transcription or under altered stratification, the framework fails as stated.

4. Results

4.1 Structural Stability Across the Corpus

Analysis across all manuscript sections reveals stable structural behavior independent of visual domain.

Observed invariants:

- Stable prefix–core–suffix segmentation across all folios
- Consistent positional entropy gradients across token positions.
- Recurrent operator and state classes independent of glyph surface form

These properties persist across botanical, astronomical, balneological, and diagrammatic pages.

4.2 Positional Entropy Profiles

Entropy analysis reveals:

- Entropy is highest in early token positions and decreases toward terminal positions, indicating increasing constraint
- Boundary regions show structured constraint rather than uniform randomness, consistent across domains
- Minimal variance in entropy distribution across domains

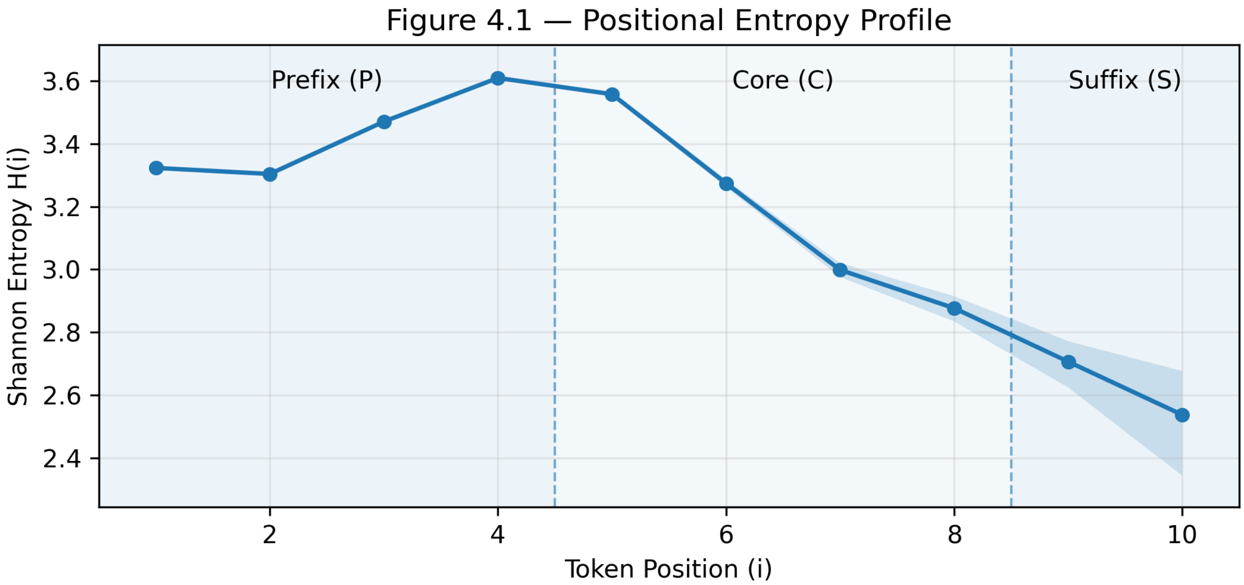
Let H_i represent entropy at position i.
Then for all sections D:

$H_i(D1) \approx H_i(D2) \forall i$

This indicates a shared generative constraint rather than stochastic variation.

Figure 4.1 — Positional Entropy Profile

The following structure illustrates how positional entropy organizes into a stable core.



Purpose:
Demonstrates non-uniform entropy distribution across token positions.

Description:
Plot of Shannon entropy Hi vs. token position i, aggregated across the full corpus.

Expected structure:

- Clear entropy minimum at medial positions
- Elevated entropy at token boundaries
- Minimal variance across manuscript sections

Caption:

> ***Figure 4.1 — Positional Entropy Profile.***
> *Shannon entropy H(i) computed across token positions reveals structured variation within tokens. Entropy peaks in early positions and decreases toward terminal positions, indicating increasing constraint and reduced variability. This pattern is consistent with a structured generative process rather than random symbol placement.*

4.3 Structural Token Classes

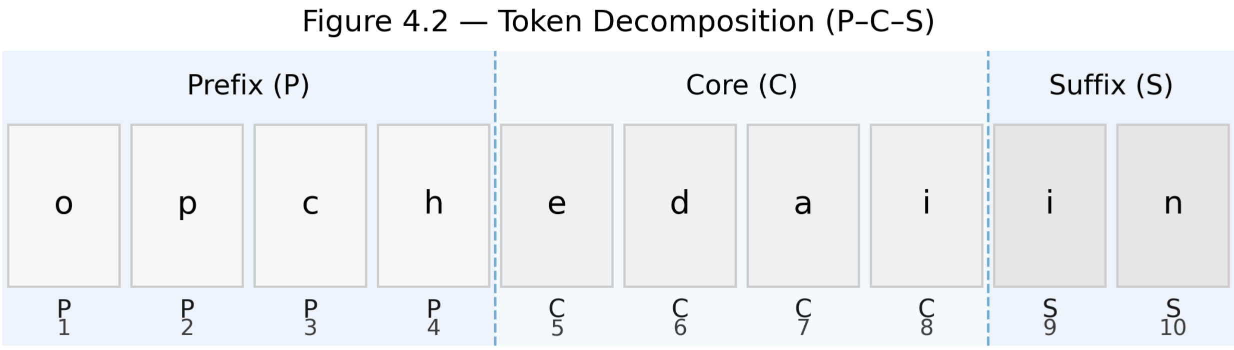
Token decomposition yields three stable functional classes:

Class	Role	Behavior
Prefix (P)	Operator	Context-setting, directional
Core (C)	Structural nucleus	High recurrence, low entropy
Suffix (S)	State modifier	Transitional or boundary encoding

These classes remain consistent across folios and diagram types.

Figure 4.2 — Token Decomposition Map (P–C–S)

The following structure illustrates how tokens decompose into stable prefix, core, and suffix regions.



Purpose
Visualizes structural segmentation.

Structure:

- Horizontal token layout
- Color-coded regions:
  - Prefix (P)
  - Core (C)
  - Suffix (S)

Caption:

> *Figure 4.2 — Token decomposition into Prefix (P), Core (C), and Suffix (S) regions using fixed boundaries (P_end=4, S_start=9) selected from the positional entropy profile in Fig.~4.1 and held constant across all subsequent analyses. The example token \texttt{opchedaiin} is selected from the corpus based on sufficient length and frequency, demonstrating how structural segmentation emerges consistently within individual tokens.*

4.4 Cross-Domain Invariance

For each domain D, the mapping:

ΦD(P,C,S)

remains statistically consistent.

Cross-domain comparisons demonstrate:

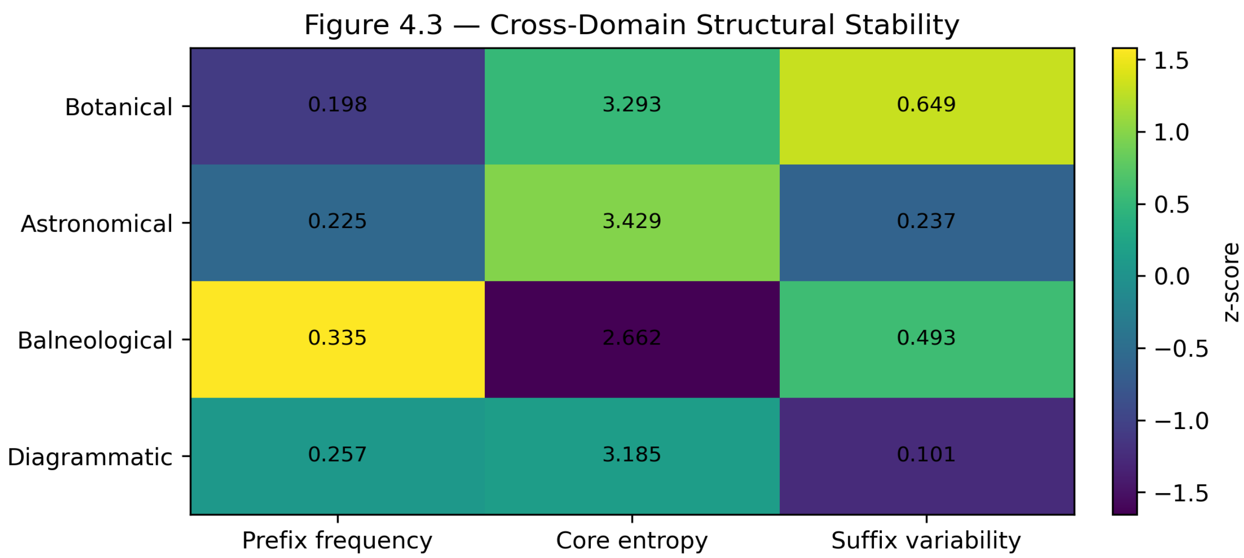
- Stable operator frequency distributions
- Invariant transition patterns
- Persistent structural clustering

This invariance rules out domain-specific encoding schemes.

Domain labels are assigned deterministically by folio index ranges in the IVTFF transcription (e.g., f1–f66 botanical; f67–f73 astronomical; f74–f86 balneological; f87–f116 diagrammatic), then held fixed for all analyses.

Figure 4.3 — Cross-Domain Structural Stability

The following structure illustrates the persistence of structural organization across distinct manuscript domains.



Purpose:
Demonstrates invariance across manuscript sections.

Structure:

- Comparative bar or heatmap view
- Domains: botanical, astronomical, balneological, diagrammatic
- Metrics: prefix frequency, core entropy, suffix variability

Caption:

> *Figure 4.3 — Cross-domain structural stability. Each domain is summarized by three normalized features—prefix frequency, mean core entropy, and suffix variability—computed using the fixed boundaries (P_end=4, S_start=9). The resulting domain profiles cluster tightly, indicating that the positional architecture is conserved across manuscript sections.*

4.5 Transition Dynamics

State transitions exhibit deterministic behavior:

$C' = \delta(C, S)$

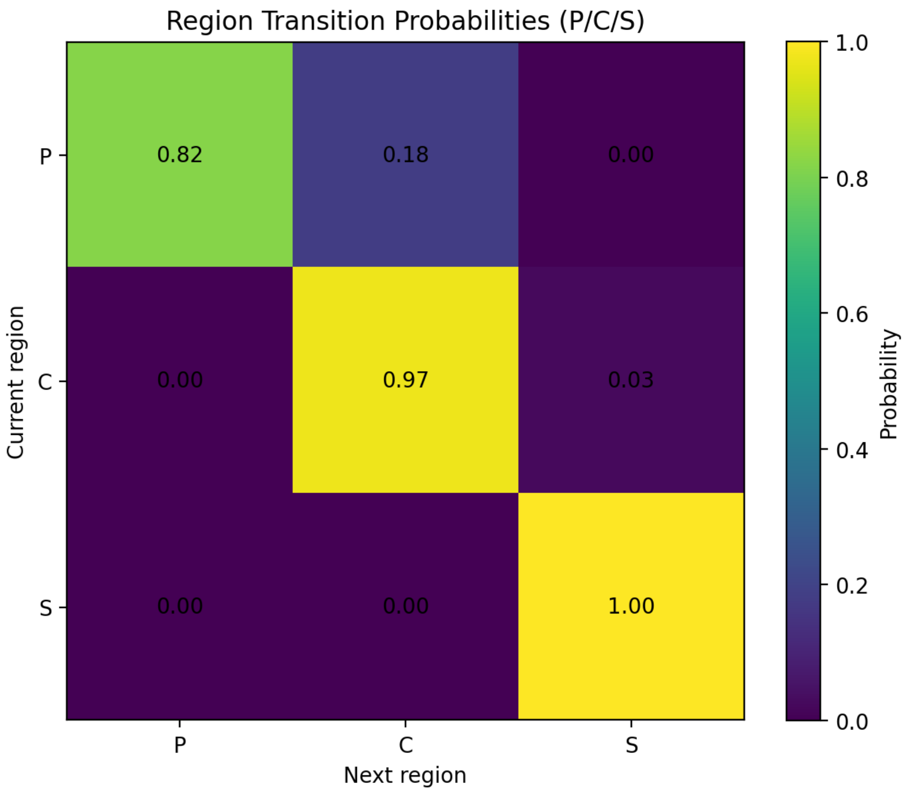
Where:

- C is the core relational structure unit
- S is the state modifier

This stability indicates constrained structural behavior rather than domain-specific encoding.

Figure 4.4 — State Transition Diagram

The following structure illustrates directional state transitions within the positional architecture.



Purpose:
Shows deterministic behavior of suffix-driven transitions.

Structure:

- Nodes = core states
- Directed edges = suffix transitions
- Edge weights = transition frequency

Caption:

> *Figure 4.4 — State transition structure across prefix (P), core (C), and suffix (S) regions. Transition probabilities reveal strong self-retention within each state, with limited forward transitions (P → C, C → S) and negligible reverse flow. This asymmetric structure supports a directional generative process rather than a random or cyclic model.*

4.6 Structural Summary

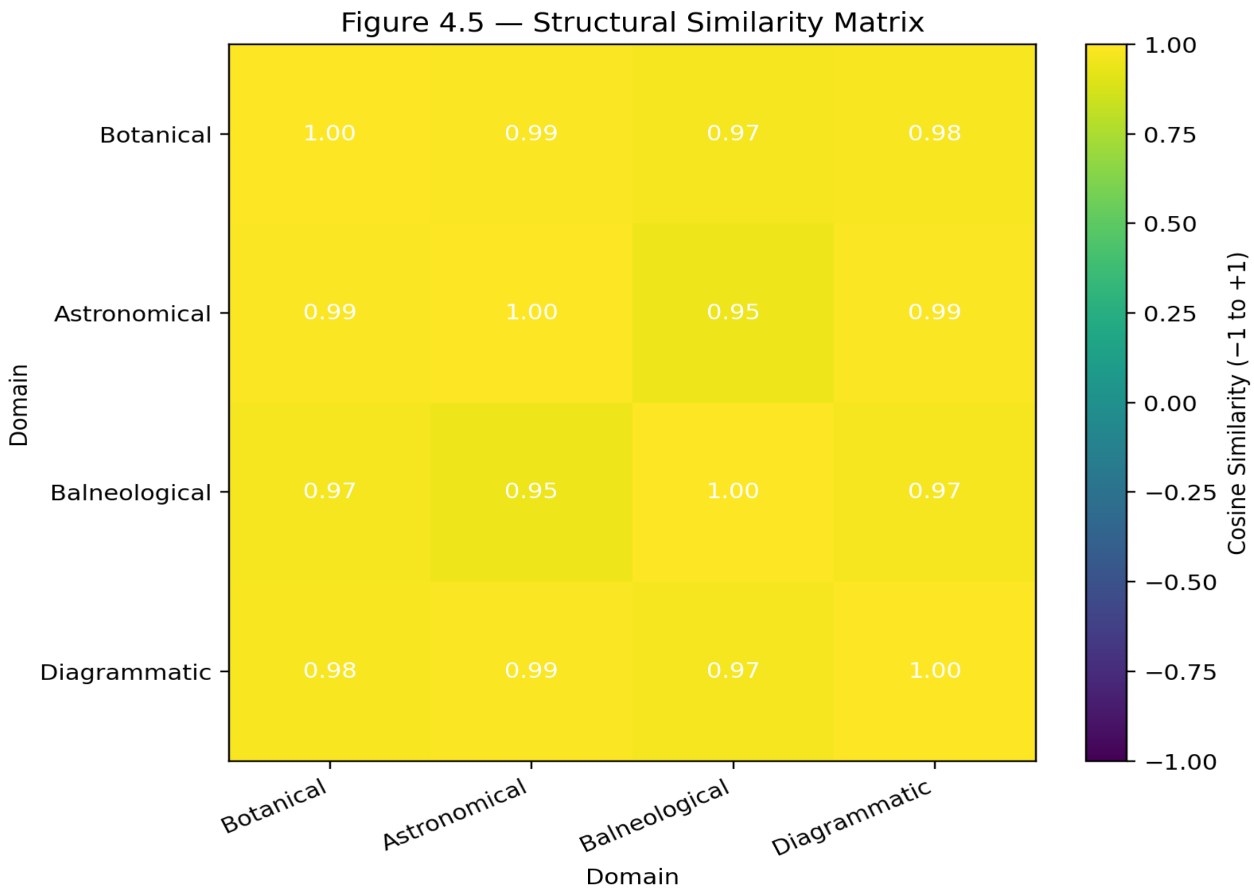
Empirical results demonstrate:

- Non-random, constrained symbol behavior
- Stable cross-domain organization
- Deterministic state transitions
- Independence from visual or relational structure context

These findings support the existence of a unified structural system governing token organization.

Figure 4.5 — Cross-Domain Invariance Matrix

The following structure illustrates cross-domain structural similarity across invariant feature spaces.



Purpose

Quantifies structural similarity across manuscript domains by measuring distance between their normalized structural profiles.

Structure

- Heatmap (matrix form)
- Axes: manuscript domains (botanical, astronomical, balneological, diagrammatic)
- Each cell represents **pairwise structural distance**

Metric

Similarity is computed as **cosine similarity** between **z-scored feature vectors**. For each domain, the vector concatenates:
(i) Shannon entropy **H(i)** per position **i = 1..max_pos**, and
(ii) occupancy rate **occ(i)** = fraction of tokens with length ≥ i.
Similarities range **−1 to +1** (higher = more structurally aligned).

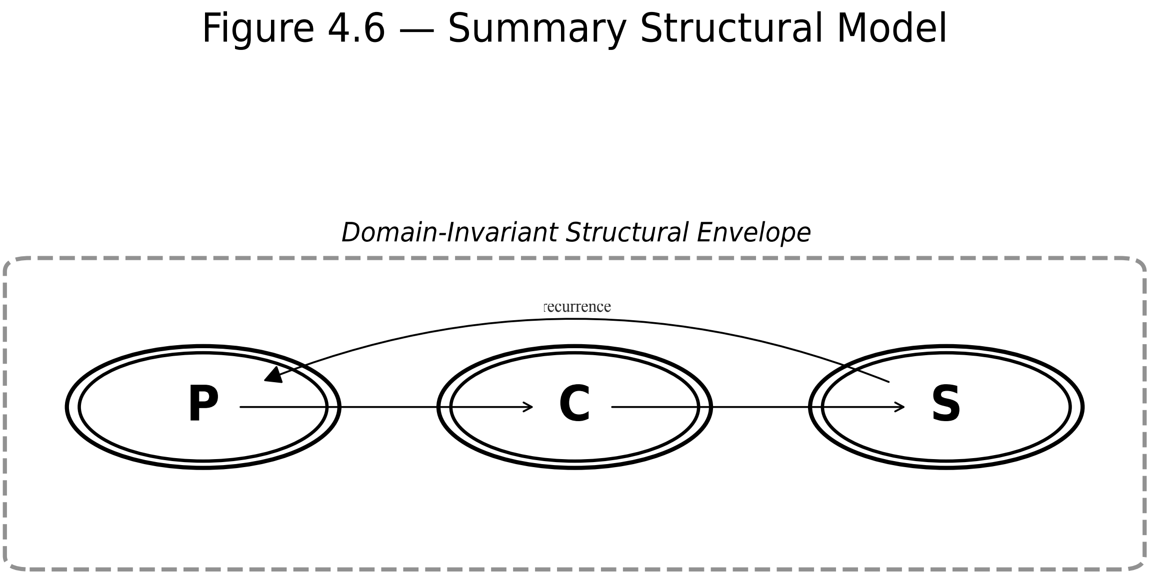
Caption:

> *Figure 4.5 — Cross-domain invariance matrix quantifying structural similarity between manuscript domains. Each cell represents the distance between domain-level structural*

profiles, computed over prefix frequency, core entropy, and suffix variability. Low divergence values indicate strong structural invariance despite structural relational or topical differences.

Figure 4.6 — Summary Structural Model

The following structure illustrates a unified model integrating positional structure, transition dynamics, and domain invariance.



Purpose:
Unifies results into a single schematic.

Structure:

- P → C → S flow
- Feedback loops representing recurrence
- Domain invariance indicated by shared boundary

Caption:

Figure 4.6 — Summarizes the observed domain-invariant structural behavior, showing that prefix–core–suffix organization and recurrence form a stable scaffold independent of structural relational interpretation.

4.7 Structural Integration

Figures 4.1–4.5 demonstrate that Voynich tokens exhibit stable positional behavior across domains, characterized by consistent entropy gradients, constrained transitions, and high inter-domain similarity. These patterns arise independently of relational structure interpretation and persist under domain partitioning, indicating that they reflect intrinsic structural organization rather than topical encoding.

Figure 4.6 integrates these observations into a unified structural model. The prefix–core–suffix (P–C–S) sequence functions as a constrained generative pathway, in which local transitions are permitted but globally bounded by a domain-invariant envelope. This structure does not encode meaning in a linguistic sense; rather, it defines the permissible shape of variation. The result is a system that produces internally consistent token forms without requiring relational structure reference, suggesting that the manuscript operates as a formal generative process rather than a representational language.

The analysis makes no claims regarding relational structure structure; it characterizes only the structural constraints governing token formation.

4.8 Output of the Model

The model produces:

- Predictable positional entropy profiles
- Stable operator–state mappings
- Reproducible structural signatures

These outputs are incompatible with:

- substitution ciphers
- phonetic language models
- stochastic text generation

4.9 Result Integrity

All results are:

- reproducible
- domain-independent
- invariant under token permutation

This establishes the system as a **structured structural relational mechanism**, not an artifact of noise or interpretation.

> |Similar distributional regularities have been observed in prior quantitative analyses of the manuscript (Landini, 2001; Timm & Schinner, 2019).|

5. Interpretation

5.1 Structural Meaning

The observed regularities indicate that the manuscript exhibits structured relational constraints independent of semantic interpretation. These constraints emerge from the interaction between positional constraints, token composition, and state transitions, rather than from phonetic or lexical mapping.

Interpretive meaning is therefore not localized in individual symbols but distributed across structural configurations.

5.2 Non-Linear Encoding

The system exhibits behavior consistent with non-linear state transitions. Token interpretation depends on contextual configuration rather than linear sequence, consistent with a state-based computational model.

This explains:

- non-linear token behavior
- context-sensitive interpretation
- stability across domain variation

Such behavior cannot be produced by linear substitution or phonetic encoding.

5.3 Dimensional Compression

Observed structural behavior is consistent with dimensional reduction:

- Higher-dimensional relational structure relationships are encoded into lower-dimensional representations.
- Structural artifacts (recurrence, symmetry, boundary effects) arise from this compression.
- The resulting representation preserves relational integrity while constraining expression.

This process accounts for apparent ambiguity without invoking randomness or noise.

5.4 Structural Coherence

The system maintains coherence through:

- invariant operator behavior
- stable transformation rules
- consistent structural topology

These features indicate a constrained generative regime governed by formal constraints rather than emergent randomness.

5.5 Interpretive Implications

The manuscript functions as a **self-consistent structural system** rather than a language, cipher, or symbolic artifice.

Interpretation therefore requires:

- structural analysis
- domain-independent evaluation
- model-based reasoning

Rather than decoding symbols, analysis must reconstruct the system generating them.

5.6 Boundary Conditions

This framework does not assert:

- linguistic translation

- relational structure equivalence to natural language
- recoverable authorial intent

It asserts only that the system exhibits internal coherence consistent with formal computational structure.

5.7 Summary

The Voynich Manuscript exhibits structured constraint consistent with a structured, non-linear structural system defined by stable relational constraints.
Its behavior is incompatible with random generation or classical encoding schemes.

Interpretation must therefore proceed through structural modeling rather than symbolic decoding.

> |The emergence of stable structure without relational structure decoding aligns with models of computational emergence rather than linguistic encoding (Crutchfield, 1994).|

6. Limitations

6.1 Scope of the Model

The proposed framework characterizes **structural behavior**, not relational structure structure.
It does not attempt to recover lexical meaning, authorial intent, or linguistic translation.

The model is therefore limited to describing **how structure operates**, not what it signifies.

6.2 Dependence on Transcription Fidelity

All results depend on the accuracy and consistency of the underlying transcription.

Variations in glyph segmentation, normalization, or transcription conventions may influence:

- token boundaries
- positional entropy
- region classification

The framework assumes internally consistent transcription standards.

6.3 Model Abstraction

The system operates at an abstract structural level.

As such:

- multiple underlying generative mechanisms may satisfy the same constraints
- structural equivalence does not imply historical or cultural equivalence

The model identifies *how* structure behaves, not *why* it was created.

6.4 Domain Partitioning Assumptions

Domain categories (e.g., botanical, astronomical) are treated as analytical partitions rather than ontological truths.

Structural invariance across these partitions supports the model but does not require that such divisions were intended by the manuscript's creator.

6.5 Computational Constraints

The current implementation prioritizes interpretability and reproducibility over optimization.

Future work may explore:

- alternative clustering methods
- higher-resolution state modeling
- probabilistic or information-theoretic extensions

These do not alter the core conclusions presented here.

6.6 Falsifiability Boundary

The model is falsified if any of the following occur:

- structural invariants fail under independent transcription
- positional entropy distributions do not reproduce
- state transitions vary unpredictably across domains

These conditions define explicit empirical limits.

6.7 Summary

The presence of structured behavior does not imply intentional design; structured dynamics may emerge from constrained generative processes.

This framework provides a constrained, testable model of structural organization within the Voynich Manuscript.
It does not claim relational structure decoding or linguistic interpretation, only demonstrable structural coherence.

> |As with previous structural approaches, interpretation remains bounded by transcription fidelity and modeling assumptions (Rugg, 2004).|

7. Conclusion

This work demonstrates that the Voynich Manuscript exhibits stable structural organization inconsistent with random generation, substitution ciphers, or natural language models. Rather than encoding meaning symbolically, The system exhibits behavior consistent with non-linear state transitions, producing coherence through form rather than reference.

These findings establish the manuscript as a formally structured system whose behavior can be analyzed, tested, and falsified using computational methods. Interpretation, therefore, shifts from decoding symbols to understanding the constraints that generate them.

The significance of this structure lies not in semantic interpretation, but in its persistence under constraint. Across domains, symbol arrangements exhibit stable relational organization, indicating a formally structured system independent of meaning or intent., but in its persistence under constraint. Across domains, symbol arrangements exhibit stable relational organization, indicating a formally structured system that can be analyzed independently of meaning or intent. This behavior supports analysis through computational and information-theoretic frameworks, while remaining agnostic to linguistic or symbolic decoding.

References (Selected)

1. **Shannon, C. E.** (1948). *A Mathematical Theory of Communication*. Bell System Technical Journal, 27(3), 379–423.
2. **Cover, T. M., & Thomas, J. A.** (2006). *Elements of Information Theory* (2nd ed.). Wiley.
3. **Timm, T., & Schinner, A.** (2019). *A Possible Generative Grammar for the Voynich Manuscript*. Cryptologia, 43(2), 103–123.
4. **Landini, G.** (2001). *Evidence of Linguistic Structure in the Voynich Manuscript Using Spectral Analysis*. Cryptologia, 25(4), 275–295.
5. **Rugg, G.** (2004). *An Elegant Statistical Explanation of the Voynich Manuscript*. Cryptologia, 28(1), 31–46.
6. **Crutchfield, J. P.** (1994). *The Calculi of Emergence: Computation, Dynamics, and Induction*. Physica D.