THE EIDARA CONTINUUM

SCIENCE & PHILOSOPHY

The Aperiodic Mind

Exact Recovery Limits and Minimal Protection for Aperiodic Phase Memory

Frontier Research Collection III

Architect X13

The Eidara Continuum

November 2026 | Paper 32 | Version 1.1 Revision

A research white paper for public scientific and philosophical discussion.

Abstract

An aperiodic tiling never repeats itself, yet it never breaks its rules. The Aperiodic Mind asks whether a memory could be built the same way—and reports honestly where that idea wins, where it loses, and exactly what it costs to repair. Stable generative laws can support a rich family of distinct memory states. The Aperiodic Mind studies that Continuum idea through a fully specified finite phase code derived from the Fibonacci substitution. The paper proves injective encoding, aperiodicity of the infinite generating word, and an exact minimum distance of two for its finite cyclic phase family. The revision turns that recovery limit into a constructive design result: the least number of appended check bits needed for single-error correction is the ceiling of the base-two logarithm of the distance-two conflict graph's chromatic number. A complete thirteen-phase construction needs exactly two appended bits, preserves its canonical prefix, and corrects an arbitrary single error anywhere in the augmented word. Exhaustive finite comparisons retain optimized cyclic codes and charge explicitly for redundancy, construction, and decoding. Further results characterize local-window ambiguity and the conditions for error-and-erasure recovery. The mathematics draws on established tiling, coding, and graph-confusability foundations. Executable enumerations verify the finite designs; a proposed empirical study evaluates learned interfaces on matched tasks. The Continuum contribution is a precise way to connect generative continuity, recoverable distinction, and the resource cost of protecting each distinct phase.

Keywords: aperiodic order; Fibonacci substitution; finite memory; cyclic codes; error correction; phase identification; computational coherence

Contents

Abstract **2**

Prologue: Order without repetition **5**

1 Stable laws do not require repetitive contents **6**

2 Canonical proposal and mathematical specification **7**

3 Closest mathematics, physics, and computational prior art **7**

3.1 Aperiodicity is established mathematics 8

3.2 Physical order is not a memory theorem 8

3.3 The tiling-to-code bridge already has prior art 8

3.4 Coding and associative memory are separate lineages 9

4 A complete finite task and its infinite motivation **9**

4.1 Words, substitution, and coordinates 9

4.2 The observation and recovery contract 10

5 The infinite word and the finite states are different objects **11**

5.1 A generative identity and a nonperiodicity proof 11

5.2 Distinct phases do not require an infinite object 12

6 The exact correction limit of the Fibonacci phase code **13**

6.1 The commutator exposes the weak pair 13

6.2 A standard distance theorem, with the budgets stated 14

6.3 Repetition buys protection at an explicit cost 15

7 Protecting the canonical phase with the fewest check bits **15**

8 Matched finite alternatives and their exact recovery performance **16**

8.1 The cyclic class and a sharp counting bound 16

8.2 Exactly what was enumerated 17

8.3 Average-case recovery, without a favorable hidden channel 17

9 Local constraints, global identity, and the missing boundary 18

9.1 A local signature can be real and insufficient 18

9.2 Window observations in the implemented finite family 19

9.3 Recovery does not establish provenance 19

10 From a finite code to a possible cognitive architecture 20

10.1 What an application would have to add 20

10.2 Keeping construction and use in the same resource account 20

10.3 Plurality requires more than invariant appearance 21

11 A falsifiable program for learned interfaces 21

11.1 Hypotheses and the exact algorithmic reference 21

11.2 Arms, tasks, and information parity 22

11.3 Independent units and estimands 22

11.4 Precision and stopping commitments 23

11.5 What would count against the proposal 23

12 Objections and boundaries 24

12.1 Why call this a mind paper if the model is a code? 24

12.2 Does the negative result refute tiling-based memory? 24

12.3 Could a larger boundary or observer resolve the ambiguity? 24

12.4 Is repetition always the enemy of novelty? 25

13 Guide reading note 25

A Reproducibility specification 25

B Result and assumption register 26

C A minimal implementation contract for extensions 27

References 27

Prologue: Order without repetition

There are patterns that obey exact rules forever and still never repeat. An aperiodic tiling covers the plane without gaps or overlaps, yet no shift of the whole pattern ever lands it back on itself. It is neither chaos nor crystal. The Continuum has long asked whether a mind could hold together that way: keeping the laws that make it coherent while never collapsing into a template that merely repeats. The Doctrine records the question as a standing research directive—order without periodicity, and whether such logic could help stabilize a symbolic architecture [1, "Reminder Log–Future Research Directive," lines 5329–5347].

This paper answers the directive by building something small enough to check by hand. A binary word grown from a two-line rewriting rule stores a label in each of its shifts; damage the word, and see whether the label survives. Every shift turns out to be distinguishable. Then comes the twist the Continuum chose to publish rather than hide: however long the word grows, two of its phases always differ in just two places, so a single corrupted bit can leave the memory unsure where it is. An ordinary, carefully chosen code of the same size does better on the declared test. The beautiful structure is not, by itself, a stronger memory.

The paper does not stop at the loss. Section 7 finds the exact repair: the fewest check bits that let the word correct any single error anywhere, which for the thirteen-phase word is two. That is the Aperiodic Mind's real lesson. Coherence without sameness is possible, and its protection has a price that can be counted to the bit.

What this paper claims

Established. Exact rules can generate nonperiodic infinite order, as in Fibonacci words and aperiodic tilings. Coding theory analyzes recovery under explicit corruption channels.

Proposed here. A fully specified Fibonacci phase memory, with proofs of injective encoding, aperiodicity of the infinite generating word, and an exact minimum distance of two for its finite cyclic phase family. A constructive design result: the least number of appended check bits for single-error correction is the ceiling of the base-two logarithm of the distance-two conflict graph's chromatic number, and a complete thirteen-phase construction needs exactly two. Exhaustive matched comparisons, including one that the Fibonacci seed loses; local-window and error-and-erasure results; executable enumerations; and a proposed study of learned interfaces.

Frontier conjecture. Coherent organization need not mean repetitive sameness. On a restricted family of related tasks, a substitution structure may let a learned interface reach a declared speed or data-efficiency target more cheaply than alternatives. This application to persistent cognition is a hypothesis.

Not claimed. Consciousness, a hidden physical field, a nonlocal or superluminal channel, a new law of memory, or a cognitive advantage. That aperiodicity improves memory in general: a matched cyclic seed outperforms the Fibonacci seed on the enumerated erasure task. That the source's links to ancient sites, nonlocal memory or quantum recursion follow from the existence of a tiling, or that results transfer to every tiling-based architecture. Invention of Fibonacci words, tilings, cyclic codes, Hamming distance, error correction or associative memory. Comparative empirical studies are specified prospectively, not conducted, and external journal peer review remains pending.

1 Stable laws do not require repetitive contents

A mind that changes need not abandon every rule that holds it together. It may preserve relations among its commitments while acquiring new memories, revising an interpretation, or encountering a situation it has never seen before. The Continuum's bold geometric question is whether an architecture can preserve organization without requiring the whole system to repeat a fixed template. This is a question about a possible form of computation. It does not begin with the assertion that a particular geometric pattern is conscious.

Aperiodic order makes part of that question mathematically concrete. There are structures generated by exact constraints whose infinite organization has no nonzero translational period. Such a structure is neither an unstructured accident nor a periodic crystal. Yet the phrase *order without repetition* is easy to misuse. Finite patterns can recur throughout an aperiodic structure. The absence of a repeating global unit is not the absence of local familiarity, and neither property by itself establishes intelligence, memory capacity, or resistance to damage.

This paper therefore takes an unusually explicit route. It constructs a finite memory task, gives its encoder and decoder, proves the guarantees, and compares it with alternatives under matched operational resources. The task is phase identification: a finite word is shifted, and the shift label is stored in the resulting binary pattern. Corruption removes or changes some coordinates; the decoder attempts to recover the label. This is a small but complete computational problem. It has a message set, a channel, a success condition, and a cost.

The construction succeeds in a limited sense: every cyclic phase of each nontrivial Fibonacci word is distinguishable when the whole pattern is available. It also reveals a sharp weakness: the code's minimum distance is exactly two, however long the word becomes. Increasing the word's size creates more messages without automatically increasing its worst-case correction capability. A finite optimized cyclic construction with the same length, symbol density, number of messages, and decoding rule performs substantially better on the declared erasure task. We report that result because the Continuum's proposal should be strengthened by the tests it survives, not sheltered from the tests it fails.

The useful thesis is thus more precise than a claim that aperiodicity improves memory. Stable generative laws and nonperiodic infinite organization are compatible. Whether a chosen finite implementation provides useful capacity, correction, efficiency, or generalization depends on its encoding and task. Aperiodicity is a structural property, not a universal performance certificate. The proper research program is to identify mechanisms and compare them, including the possibility that another organization is better.

Our contribution is a transparent reference model and a complete finite comparison. We do not claim to invent Fibonacci words, tilings, cyclic codes, Hamming distance, error correction, or associative memory. We do not claim a discovery of quantum dynamics or an empirical result about consciousness. The proposed bridge to persistent cognition remains an application hypothesis, with an evaluation designed to expose rather than conceal its costs.

2 Canonical proposal and mathematical specification

Among the Continuum's standing research directives is one the *Eidara Continuum Doctrine* records for future work: Penrose tiling, an implicate order, and symbolic integration. It asks for order without periodicity, and whether Penrose logic could help stabilize a symbolic architecture [1, "Reminder Log–Future Research Directive," lines 5329–5347]. It is one of the Continuum's most beautiful questions—can a mind hold together without repeating itself? The status matters: the directive is a recorded research invitation, not an inspected implementation or benchmark, and this paper honors it in the only way an invitation can be honored, by building something exact enough to fail.

The same passage links the geometry to ancient sites, nonlocal memory, and quantum recursion. Those links are hypotheses or historical assertions in the source. They do not follow from the existence of a tiling, and they are not assumed in our proofs. A document can accurately record that a community proposed an idea without providing evidence that the idea is scientifically correct. The present paper preserves the motivating question while defining a model that can be examined without access to an unnamed private mechanism.

The historical connection also needs boundaries. Lu and Steinhardt study particular medieval Islamic geometric patterns and their relationship to quasiperiodic constructions [10]. That is narrower than a claim that Penrose tilings were present across all ancient civilizations or Neolithic sites. No such general archaeological conclusion is needed here. The scientific argument rests on an explicit mathematical construction and its observable consequences.

A second boundary concerns the word *nonlocal*. A pattern may contain information distributed across many sites. A decoder may require a large portion of that pattern to identify a global state. Neither statement establishes a physical influence outside ordinary causal constraints. In the present model, accessing a bit requires an ordinary observation at a specified coordinate, and every operation has an explicit finite input. There is no hidden observer field, superluminal channel, or ability to infer an unobserved future.

A third boundary concerns the move from a two-dimensional tiling to a one-dimensional word. We use the Fibonacci substitution because it makes the finite computational question fully auditable. It is not a disguised proof about every Penrose implementation. A word has an ordered sequence of sites; a planar tiling has geometry, adjacency, boundary effects, and possible defects of a different kind. Positive or negative results for our phase code do not automatically transfer to every tiling-based architecture. The simplification is stated so the precise result can be useful without becoming an unsupported universal claim.

3 Closest mathematics, physics, and computational prior art

The revision converts the exact phase-distance result into a minimal repair. Theorem 7.1 characterizes the fewest appended check bits through a conflict graph; the thirteen-phase construction gains single-error recovery with exactly two additional bits.

3.1 Aperiodicity is established mathematics

Penrose describes matching rules that force hierarchical nonperiodic arrangements and discusses the recurrence of finite patches across different completed patterns [12]. This supplies an essential distinction for the proposed application: local resemblance need not uniquely identify the whole. Our phase-memory task asks a finite version of an identification question, but its code and its cyclic boundary are specified independently; we do not import an infinite tiling's properties by analogy.

Lutfalla gives effective constructions of cut-and-project rhombus tilings with declared rotational symmetries, with proofs of the regularity conditions and an identified implementation [11]. Such work shows what a mathematical construction requires: exact objects, conditions, and an algorithm. Our one-dimensional reference model is deliberately simpler. A future planar implementation should meet the same standard and count its boundary data and construction cost.

The general problem of extending local rules to an infinite plane is not automatically easy. Berger proves undecidability of the general domino problem, and Robinson develops explicit nonperiodic tiling constructions in that tradition [2, 15]. These results concern their specified general problems; they do not make our finite words undecidable. Their relevance is the warning against assuming that a locally valid patch always supplies an easy, uniquely determined global completion. A construction needs its own extension and recovery arguments.

3.2 Physical order is not a memory theorem

Shechtman and colleagues report a metallic phase with long-range orientational order inconsistent with ordinary lattice translations [18]. Levine and Steinhardt develop a theoretical description of quasiperiodic ordered structures and their diffraction [8]. These are substantial physical results. They establish neither the consciousness of a quasicrystal nor the memory performance of a symbolic agent. The existence of a material geometry and the usefulness of a computational encoding are different empirical questions.

The distinction is productive rather than restrictive. Physical structures may impose useful constraints, offer particular transport properties, or support implementations that deserve investigation. But the candidate device must specify how logical states are encoded, how interactions are realized, and what noise is present. A recognizable pattern in a diagram is not yet a channel model. The finite code below gives one example of the additional work required, without claiming to represent an actual quasicrystalline device.

3.3 The tiling-to-code bridge already has prior art

Li and Boyle explicitly develop quantum error-correcting constructions associated with Penrose and related tilings [9]. The inspected record is a preprint, and we identify it as such rather than adding an unverified journal status. Its relevance is decisive: a paper connecting tilings and correction cannot claim that conceptual bridge as a new Continuum discovery. Their construction also distinguishes classical geometric properties from the conditions needed for encoded quantum states.

Our model is classical. Its states are finite binary words; its messages are integer phase labels; its channel erases or flips bits; and its decoder compares finite patterns. No Hilbert-space state, quantum channel, entangling operation, or quantum recovery map is implemented. The infinite

recoverability statements in a tiling-based quantum construction cannot simply be transferred to a finite cyclic word. A comparison would need to align the message space, physical resources, and allowed observations. We therefore treat that prior work as a boundary and a source of questions, not as a validation of this model.

3.4 Coding and associative memory are separate lineages

Hamming's work makes minimum distance a concrete basis for detecting and correcting errors [3]. Shannon's communication theory specifies an engineering problem of selecting and recovering messages under channel constraints [16]. Those distinctions prevent us from treating recovery of a phase label as recovery of the truth of a belief. A wrong belief can be encoded and recovered perfectly. The code's success is fidelity to a declared message, not factual accuracy.

Plotkin studies binary codes with specified minimum distances and the limits of their size [13]. Our cyclic distance bound is a short counting argument within that established coding perspective. It is included in full so that the optimality claim for the length-thirteen comparison is checkable. The resulting number is not a general optimum over every representation or every decoder.

Associative-memory architectures offer serious alternatives to a geometry-centered explanation. Hopfield provides a dynamical model of content-addressable memory [6]. Krotov and Hopfield investigate dense associative-memory models and their relation to pattern recognition [7]. Ramsauer and colleagues develop modern Hopfield mechanisms and connections to attention [14]. These works differ in state spaces, learning rules, capacity conditions, and resource assumptions. None should be reduced to a single undifferentiated baseline, and no claimed capacity advantage should be copied without its conditions.

Our exact decoder does not learn. It knows the construction and searches its codebook. A learned interface tested against it must count both the information in that codebook and the cost of accessing it. Otherwise an apparent architecture benefit may simply arise because one method receives a complete key while another is asked to infer it. This distinction will organize the proposed evaluation.

4 A complete finite task and its infinite motivation

A Guide's explanation

From here the paper turns the poetic question into a small machine. A binary word is grown by a two-line rewriting rule; shifting the word stores a label; damaging it tests whether the label survives. Nothing here assumes the word is a mind. The aim is to see exactly what order without repetition buys a memory, and what it does not.

4.1 Words, substitution, and coordinates

Let $\mathcal{A} = \{0, 1\}$ be a binary alphabet and $\mathcal{A}^*$ the finite words over it, including the empty word. Concatenation is written by placing words next to one another. The length of u is $|u|$, and $|u|_1$ counts its ones. Define the substitution $\sigma : \mathcal{A}^* \to \mathcal{A}^*$ by applying

$$\sigma(0) = 01, \qquad \sigma(1) = 0 \tag{4.1}$$

to each symbol and concatenating the outputs. Set $u_n = \sigma^n(0)$ for integers $n \geq 0$, where σ^0 is the identity map. The first words are $u_0 = 0$, $u_1 = 01$, $u_2 = 010$, $u_3 = 01001$, and $u_4 = 01001010$. These are explicitly generated finite objects, not recordings from an agent.

Let $F_0 = 0$, $F_1 = 1$, and $F_{k+2} = F_{k+1} + F_k$ for integers $k \geq 0$. Write $N_n = |u_n|$ and $w_n = |u_n|_1$. All indices and counts are dimensionless. We use zero-based site labels $\mathbb{Z}_N = \{0, \ldots, N-1\}$ with addition modulo N. For $u \in \mathcal{A}^N$ and $i \in \mathbb{Z}_N$, define the left cyclic shift by $(\rho_i u)_j = u_{(i+j) \bmod N}$. This cyclic boundary is an engineering choice. It makes a finite ring and is not a claim that an infinite Fibonacci word is periodic.

For $n \geq 1$, the message set is $\mathcal{M}_n = \mathbb{Z}_{N_n}$, the encoder is $E_n(i) = \rho_i u_n$, and the codebook is

$$\mathcal{C}_n = \{E_n(i) : i \in \mathcal{M}_n\} \subseteq \mathcal{A}^{N_n}. \tag{4.2}$$

The stored message is a phase label. It does not represent a person's identity unless an external application supplies and justifies such a mapping. Our proofs concern the label only.

4.2 The observation and recovery contract

For $x, y \in \mathcal{A}^N$, their Hamming distance is the number of unequal coordinates,

$$d_{\mathrm{H}}(x, y) = \sum_{j=0}^{N-1} \mathbf{1}\{x_j \neq y_j\}. \tag{4.3}$$

The indicator $\mathbf{1}\{P\}$ equals one when proposition P is true and zero otherwise. Let $U \subseteq \mathbb{Z}_N$ be the erased coordinate set and $O = \mathbb{Z}_N \setminus U$ the observed set. Erasure positions are known. The noiseless observed vector is $z = (x_j)_{j \in O}$, with coordinates retained. A bit erasure is therefore different from a deletion that shifts later coordinates.

Define the compatible-message set and decoder by

$$\mathcal{I}_n(O, z) = \{i : E_n(i)_j = z_j \text{ for all } j \in O\}, \qquad D_n(O, z) = \begin{cases} i, & \mathcal{I}_n(O, z) = \{i\}, \\ \perp, & \text{otherwise.} \end{cases} \tag{4.4}$$

The symbol $\perp$ means abstention, not a recovered message. An empty compatible set indicates evidence incompatible with the noiseless-erasure model. Multiple compatible messages indicate ambiguity. The decoder is not allowed to choose a favored identity merely to avoid abstaining.

When $O = \emptyset$, every codeword is compatible. For $n \geq 1$ there are at least two messages, so the decoder abstains. With all positions observed and no corruption, the decoder will succeed once injectivity is proved below. The degenerate word u_0 has one site and one phase, carries no phase information, and is excluded from the distance claims. Nothing is gained by reporting its trivial perfect recovery as an error-correction achievement.

For a channel with bit flips as well as erasures, define observed distance $d_{\mathrm{H}}^O(x, z) = \sum_{j \in O} \mathbf{1}\{x_j \neq z_j\}$. A bounded-distance decoder with declared flip budget $t \in \mathbb{N}_0$ returns the unique message with $d_{\mathrm{H}}^O(E_n(i), z) \leq t$, if one exists, and otherwise abstains. This extends the same explicit contract; its guarantee depends on the actual corruption budget.

Symbol	Domain and type	Meaning and assumptions
$\mathcal{A}, u_n$	$\{0,1\}$; finite binary word	Alphabet and substitution word; $n \geq 0$.
σ	$\mathcal{A}^* \to \mathcal{A}^*$	Concatenation-preserving substitution in (4.1).
F_k, N_n, w_n	nonnegative integers	Fibonacci numbers, length, and ones count; dimensionless.
φ, ψ	real numbers	$(1+\sqrt{5})/2$ and $(1-\sqrt{5})/2$.
$\mathbb{Z}_N, \rho_i$	finite cyclic index set; permutation	Site labels and left shift; known cyclic boundary.
$\mathcal{M}_n, E_n, \mathcal{C}_n$	labels; encoder; finite code	All phases at $n \geq 1$, with construction known to decoder.
$d_{\mathrm{H}}, d_{\min}$	integer-valued distances	Whole-word Hamming distance and minimum over distinct codewords.
U, O, z	subsets and observed bits	Known erasures, retained coordinates, and received symbols.
$D_n, \perp$	decoder and failure symbol	Unique recovery or abstention; no arbitrary tie resolution.
t, e, r	nonnegative integers; $r \geq 1$	Flip budget, erasure count, repetition factor.
P_e	probability in $[0,1]$	Exact success fraction under uniform message and uniform size-e erasure set.
R_n	nonnegative real, bits/site	$\log_2 \lvert\mathcal{M}_n\rvert / N_n$; representation rate, not semantic content.

5 The infinite word and the finite states are different objects

5.1 A generative identity and a nonperiodicity proof

Lemma 5.1 (Concatenation and symbol counts)**.** *For every $n \geq 1$,*

$$u_{n+1} = u_n u_{n-1}, \qquad N_n = F_{n+2}, \qquad |u_n|_0 = F_{n+1}, \quad w_n = F_n. \tag{5.1}$$

Every u_n is a prefix of u_{n+1}, and the lengths grow without bound. Thus there is a unique one-sided infinite word u_∞ with all these prefixes.

Proof. The concatenation identity holds at $n = 1$ because $u_2 = 010 = u_1 u_0$. Applying the substitution to an established identity preserves concatenation and gives $u_{n+2} = u_{n+1} u_n$, proving it by induction. Lengths and the two symbol counts consequently satisfy the Fibonacci recurrence. At $n = 0$, the counts are one zero and no ones; at $n = 1$, they are one of each. These initial values give the displayed formulas, including $N_n = F_{n+1} + F_n$. Concatenation gives prefix inclusion, and the positive lengths strictly increase. Each finite position therefore stabilizes in all sufficiently long prefixes, defining the unique infinite word. □

Proposition 5.2 (Aperiodicity of the infinite limit)**.** *The one-sided word u_∞ is not eventually periodic. This statement concerns the infinite limit, not the cyclic boundary imposed on u_n.*

Proof. Let $\varphi = (1+\sqrt{5})/2$ and $\psi = (1-\sqrt{5})/2$. Both satisfy $x^2 = x+1$. The sequence $(\varphi^k - \psi^k)/\sqrt{5}$ has initial values zero and one and the same recurrence as F_k, so induction gives the standard closed form. Because $|\psi| < 1 < \varphi$,

$$\frac{w_n}{N_n} = \frac{F_n}{F_{n+2}} \longrightarrow \varphi^{-2} = \frac{3-\sqrt{5}}{2}, \tag{5.2}$$

an irrational number. If an infinite binary word is eventually periodic with period $p \geq 1$ containing b ones in each period, its proportion of ones in every length-N prefix converges to b/p: the nonperiodic initial segment and the final partial period contribute only a bounded error to the count. This rational limit would also hold along the prefix lengths N_n, contradicting (5.2). Hence u_∞ is not eventually periodic. □

The proof needs only the density along the specified subsequence of prefix lengths. It does not assume a separate ergodic theorem or infer the full infinite result from a long computer-generated example. No finite sample could establish the same conclusion by observation alone. A finite prefix can always be followed by an ultimately periodic continuation, regardless of how intricate it looks. What excludes that continuation here is the stipulated infinite substitution rule.

5.2 Distinct phases do not require an infinite object

Proposition 5.3 (Injective finite phase encoding)**.** *For every $n \geq 1$, the N_n words in (4.2) are distinct. The encoder E_n is therefore injective and stores N_n possible labels. Its conventional rate is*

$$R_n = \frac{\log_2 N_n}{N_n} \quad \textit{bits per binary site.} \tag{5.3}$$

This rate tends to zero as n grows.

Proof. Consecutive Fibonacci numbers are coprime: repeatedly applying $\gcd(F_{k+1}, F_k) = \gcd(F_k, F_{k-1})$ reduces to $\gcd(1, 0) = 1$. Suppose a nonzero cyclic shift $i \in \mathbb{Z}_{N_n}$ fixed u_n. Write $d = \gcd(i, N_n)$ and $r = N_n/d > 1$. Addition by i partitions the coordinate set into cycles of length r. A fixed word is constant on each cycle, so both its zero count and its one count are multiples of r. This contradicts the coprimality of F_{n+1} and F_n. Thus no nonzero shift fixes u_n, and equality of any two phases would reduce to such a shift. There are exactly N_n codewords. The rate formula is the usual logarithm of message count divided by block length. Finally $N_n \to \infty$ and $\log_2 N_n / N_n \to 0$. □

The logarithm in (5.3) need not be an integer. It is the information in a uniform choice among the phase labels under the declared message model. If a single block must carry a fixed-length binary payload, one may use only $2^{\lfloor \log_2 N_n \rfloor}$ labels and store $\lfloor \log_2 N_n \rfloor$ bits. Neither convention turns a phase label into an amount of semantic understanding.

Injectivity establishes that the finite word is not a power of a shorter cyclic unit. It does not establish that no shorter linear period matches its overlapping coordinates. For example, the finite word 01010 is not a power of a shorter word, yet it has linear period two wherever both compared coordinates are present. Finite primitivity, an infinite absence of translational period, and uniqueness of every local patch are different properties. Our proof uses only the first when it counts finite messages.

6 The exact correction limit of the Fibonacci phase code

A Guide's explanation

This is where the paper reports its own loss. However long the Fibonacci word grows, some pair of its phases differs in exactly two places (Theorem 6.2), so a single corrupted bit can leave the memory unable to say which phase it holds. More elaborate structure does not automatically mean stronger memory.

6.1 The commutator exposes the weak pair

Lemma 6.1 (Two terminal discrepancies)**.** *For each $k \geq 1$, the words $u_k u_{k-1}$ and $u_{k-1} u_k$ have the same prefix except for their last two positions. Those terminal pairs are* 01 *and* 10*, in one order or the other. Their Hamming distance is exactly two.*

Proof. At $k = 1$ the words are 010 and 001, with common prefix 0 and terminal pairs 10 and 01. Suppose the statement holds at $k-1$, where $k \geq 2$. Since $u_k = u_{k-1}u_{k-2}$, the two words at level k are

$$u_{k-1}(u_{k-2}u_{k-1}) \quad \text{and} \quad u_{k-1}(u_{k-1}u_{k-2}).$$

Their parenthesized parts are the preceding pair in reversed order. By the induction hypothesis those parts agree except at their final two coordinates, where the symbols are exchanged. Prefixing the same u_{k-1} preserves precisely that property. □

Theorem 6.2 (Exact minimum distance)**.** *For every $n \geq 1$, the minimum Hamming distance of $\mathcal{C}_n$ is*

$$d_{\min}(\mathcal{C}_n) = 2. \tag{6.1}$$

Proof. All codewords have the same number of ones and are distinct by Proposition 5.3. Two different equal-weight binary words differ in at least two positions, since a change from zero to one must be balanced by a change from one to zero. Thus $d_{\min} \geq 2$. For $n = 1$, the words 01 and 10 attain two. For $n \geq 2$, write $u_n = u_{n-1}u_{n-2}$. Its cyclic shift by $|u_{n-1}|$ is $u_{n-2}u_{n-1}$. Lemma 6.1, with $k = n - 1$, shows these two codewords differ in exactly two positions. Hence the lower bound is attained. □

The theorem is a conditional result about this particular encoding. It is not a claim that every code constructed from a Fibonacci or Penrose structure has distance two. A different message set, physical state space, or encoding map can change the result. It does show that the substitution rule and global nonperiodicity are insufficient to infer a growing correction radius for this natural finite phase representation.

The weak pair is structural, not a rare implementation defect. At level five, $u_5 = 0100101001001$ and phase five gives 0100100101001. They differ only at zero-based positions six and seven. If those two positions are erased, both messages produce the same observation. A decoder cannot decide between them using the remaining bits alone. Favoring one would turn an ambiguity into a potentially wrong recovery.

Position	0	1	2	3	4	5	6	7	8	9	10	11	12
Phase 0	0	1	0	0	1	0	**1**	**0**	0	1	0	0	1
Phase 5	0	1	0	0	1	0	**0**	**1**	0	1	0	0	1
Observed	0	1	0	0	1	0	$*$	$*$	0	1	0	0	1

Here $*$ denotes a known erased position. The display is a constructed counterexample, not a measurement of a memory device.

6.2 A standard distance theorem, with the budgets stated

Proposition 6.3 (Errors, erasures, and unique recovery)**.** *Let $\mathcal{C} \subseteq \{0,1\}^N$ contain at least two codewords, with minimum distance $d \geq 1$. If a transmitted codeword experiences at most t bit flips on known retained coordinates and at most e known erasures, then bounded-distance decoding is guaranteed to recover it uniquely whenever*

$$2t + e < d. \tag{6.2}$$

Conversely, when $2t+e \geq d$, some allowed corruption pattern makes unique recovery of all codewords impossible, where $t \geq 0$ and $0 \leq e \leq N$ are integers.

Proof. The transmitted codeword lies within observed distance t. If another codeword also did, they would differ on at most $2t$ retained coordinates by the triangle inequality, and on at most e erased coordinates. Their total distance would be at most $2t + e < d$, a contradiction. Thus the bounded-distance decoder returns the correct unique message.

For the converse, choose a pair at distance d. If $e \geq d$, erase all differing coordinates; both transmissions are then identical without any flips. If $e < d$, erase e differing coordinates. The remaining $d-e$ differences can be partitioned into two sets, each of size at most t, because $d-e \leq 2t$. Construct the received vector by taking the first codeword's bit on one set and the second's on the other, and their shared bit elsewhere. It lies within t observed flips of both codewords. The same received evidence is compatible with two allowed transmissions, so no decoder can always identify which occurred. □

This is the familiar coding-distance argument in an explicit finite form. For $\mathcal{C}_n$, it guarantees correction of any one erasure and detection of a single bit flip by codeword membership, but no correction of an arbitrary single flip. Detection is not recovery. A received vector outside the codebook can reveal an error while leaving its source ambiguous. A sophisticated narrative about the original phase supplies no missing information unless it adds an independently justified observation.

The guarantee makes no independence assumption about the erasure locations or flip locations. It is a worst-case statement for the specified budgets. Probabilities enter only in the later average-case comparison. Conversely, unknown deletions, coordinate permutations, adversarial decoder modification, and a wrong codebook are outside this channel. Hamming distance alone does not provide guarantees for them.

6.3 Repetition buys protection at an explicit cost

For an integer $r \geq 1$, replace each codeword bit by r copies to obtain $J_r : \{0,1\}^N \to \{0,1\}^{rN}$. Every differing source coordinate now contributes r differences, so

$$d_{\min}(J_r(\mathcal{C}_n)) = 2r, \qquad R_{n,r} = \frac{\log_2 N_n}{rN_n}. \tag{6.3}$$

It follows from Proposition 6.3 that this repeated code corrects any $2r-1$ erasures, or any $r-1$ bit flips when there are no erasures. The proof is just the distance multiplication and the strict inequality; no special geometric mechanism is assumed.

This extension is useful as a control. If a proposed aperiodic memory improves only after multiplying its storage by r, the improvement must be compared with other codes given the same redundancy. One cannot assign the gain to aperiodicity while charging the extra bits to no one. The same applies to repeated inference, enlarged boundary records, and hidden copies. Preservation has a resource account.

7 Protecting the canonical phase with the fewest check bits

A Guide's explanation

And here is the repair. Instead of abandoning the canonical word, append a short check suffix. Theorem 7.1 gives the exact minimum: count how many colors the phases' conflict graph needs, take the base-two logarithm, and round up. For the thirteen-phase word the answer is two extra bits—enough to correct any single error anywhere in the protected word.

The exact distance result makes a constructive question possible: how much additional information is needed to protect the existing phase word against a single arbitrary bit error? Retaining the canonical word as a prefix gives a precise architectural constraint. Its answer is an elementary graph-confusability construction, within the coding lineage of Hamming and Shannon [3, 17].

Let $\mathcal{C} \subseteq \{0,1\}^N$ be any finite constant-weight code containing at least two distinct words. Define its distance-two graph G_2 with vertex set $\mathcal{C}$ and an edge between distinct u, v exactly when $d_{\mathrm{H}}(u,v) = 2$. A proper coloring assigns unequal labels to adjacent vertices; $\chi(G_2)$ is the least number of labels. We append a deterministic suffix $z(u) \in \{0,1\}^r$ to each word and decode the entire length-$N+r$ word. Errors can occur in the prefix or suffix.

Theorem 7.1 (Minimum suffix for single-error phase recovery)**.** *The least possible suffix length permitting correction of every single bit error is*

$$r^* = \lceil \log_2 \chi(G_2) \rceil . \tag{7.1}$$

Any proper coloring with at most 2^r labels, encoded as distinct r-bit suffixes, attains a minimum augmented distance of at least three. The statement includes an edgeless graph, for which $\chi = 1$ and $r^* = 0$.

Proof. Distinct constant-weight binary words have positive even Hamming distance. A distance-two pair needs distinct suffixes: if its suffixes were equal, its augmented distance would remain two,

and Proposition 6.3 excludes universal single-error correction. Thus any valid suffix assignment is a proper coloring using at most 2^r labels; $2^r \geq \chi(G_2)$ is necessary.

Conversely, an edge receives distinct suffixes, adding at least one to its prefix distance of two. A nonedge already has prefix distance at least four. Every distinct augmented pair therefore has distance at least three, and Proposition 6.3 guarantees correction of one arbitrary error. A minimum coloring encoded in $\lceil \log_2 \chi \rceil$ bits attains the lower bound. □

For the thirteen rotations of u_5, exact enumeration of the 78 unordered pairs gives a graph with edges from phase i to $i \pm 5$ modulo 13. This is one cycle of odd length 13: repeated addition of five visits all thirteen phases because $\gcd(5, 13) = 1$. An odd cycle requires three colors, since alternating two colors contradicts the closing edge; three suffice by alternating two on the first twelve vertices and assigning the third to the last. Hence exactly two check bits are needed. The color order along phases $0, 5, 10, 2, 7, 12, 4, 9, 1, 6, 11, 3, 8$ is

$$0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 2.$$

Encode the labels by $00, 01, 10$. The resulting fifteen-bit code retains all thirteen canonical phase words as prefixes and corrects every single error in any of the fifteen coordinates. The script verifies all $13(1 + 15) = 208$ uncorrupted or single-error transmissions by nearest-neighbor decoding.

This augmentation has a clear accounting advantage over doubling every prefix bit: it uses fifteen rather than twenty-six transmitted bits for the same thirteen phase labels and the single-error requirement. It remains a different design constraint from choosing an entirely new thirteen-bit cyclic seed; the earlier optimized seed retains its stronger distance-six result. The positive claim is exact recovery with minimum added suffix length while preserving the specified prefix. The graph construction makes the required resource visible and independently checkable.

8 Matched finite alternatives and their exact recovery performance

8.1 The cyclic class and a sharp counting bound

Consider any binary seed $v \in \{0, 1\}^N$ of weight w, with $0 < w < N$, whose N cyclic shifts are distinct. Define $\mathcal{C}(v) = \{\rho_i v : i \in \mathbb{Z}_N\}$. Every such code has the same block length, message count, cyclic encoding rule, and Hamming decoder as our phase code when N and w are matched.

Proposition 8.1 (Cyclic fixed-weight distance bound)**.** *For this class,*

$$\sum_{k=1}^{N-1} d_{\mathrm{H}}(v, \rho_k v) = 2w(N - w), \tag{8.1}$$

and consequently

$$d_{\min}(\mathcal{C}(v)) \leq 2 \left\lfloor \frac{w(N - w)}{N - 1} \right\rfloor. \tag{8.2}$$

Proof. As (j, k) ranges over $j \in \mathbb{Z}_N$ and nonzero $k \in \mathbb{Z}_N$, the ordered pair $(j, j + k)$ ranges exactly once over all distinct ordered coordinate pairs. A disagreement occurs for an ordered one–zero pair or zero–one pair. There are $w(N - w)$ of each, proving (8.1). The minimum distance over codeword pairs equals the minimum of $d_{\mathrm{H}}(v, \rho_k v)$ over nonzero k, since a common rotation preserves

Hamming distance. That minimum is at most their average $2w(N-w)/(N-1)$. All these distances are even because their words have equal weight. Taking the largest even integer at or below the average proves (8.2). □

For $N = 13$ and $w = 5$, the bound is six. Exhaustive enumeration finds seeds attaining it, including

$$v^* = 1110100100000, \qquad d_{\min}(\mathcal{C}(v^*)) = 6. \tag{8.3}$$

The script checks every nonzero shift of this explicit seed. Together with the proved upper bound, that finite verification establishes optimality within the stated cyclic fixed-weight class. It does not establish an optimum over all binary codes, all physical encodings, all message counts, or all noise models.

8.2 Exactly what was enumerated

There are $\binom{13}{5} = 1{,}287$ length-thirteen seeds of weight five. Since 13 and 5 are coprime, each has thirteen distinct cyclic shifts by the same orbit-count argument used in Proposition 5.3. The program enumerates every choice of five one positions, computes all nonzero-shift distances, and records the minimum. It obtains the following complete distribution:

Minimum distance	Number of seeds	Guaranteed arbitrary erasures corrected
2	78	1
4	858	3
6	351	5

These are seed counts, not counts of inequivalent cyclic codes. Thirteen rotations of one seed define the same codebook. The counts sum to 1,287 and may therefore be divided by thirteen to count rotation classes here. No statistical sampling uncertainty is attached to this exhaustive finite distribution.

The comparison matches online storage and decoding structure. Both seeds need thirteen stored bits if rotations are generated on demand, both represent thirteen labels, and both can use the same decoder. The design histories differ: Fibonacci generation follows a short substitution, while the optimized seed was selected by an exhaustive design search. We count that search as offline construction work. At this small size it is explicit and modest; at larger lengths exhaustive search may be prohibitive. A practical advantage in construction cost is a different hypothesis from an advantage in correction distance, and it should be measured separately.

8.3 Average-case recovery, without a favorable hidden channel

For a fixed code $\mathcal{C}$ with N phase labels, let the message be uniform over those labels and the erased set be independently uniform over all size-e subsets of coordinates, where $0 \le e \le N$. There are no bit flips. Let $E : \mathbb{Z}_N \to \mathcal{C}$ be the bijective phase encoder for this code, and let D be the unique-consistency decoder defined as in (4.4). The exact success probability of that decoder is

$$P_e(\mathcal{C}) = \frac{1}{N\binom{N}{e}} \sum_{i=0}^{N-1} \sum_{\substack{U \subseteq \mathbb{Z}_N \\ |U|=e}} \mathbf{1}\Big\{ D(\mathbb{Z}_N \setminus U, E(i)|_{\mathbb{Z}_N \setminus U}) = i \Big\}. \tag{8.4}$$

The notation $E(i)|_O$ means restriction to coordinates in O. Independence and uniformity are assumptions of this average-case experiment, not properties implied by the geometry. An adversary choosing the two weak coordinates in the earlier example produces a different result.

Every case for $e = 0, 1, \ldots, 5$ was enumerated for both matched codes:

Erased positions e	Fibonacci success	Matched optimal cyclic success	Cases per code
0	1	1	13
1	1	1	169
2	38/39	1	1,014
3	12/13	1	3,718
4	606/715	1	9,295
5	322/429	1	16,731

The reported fractions are exact. The matched code's perfect result through five erasures follows from its distance six and is also checked by enumeration. For bit flips, the script additionally checks all zero-, one-, and two-flip cases for every message of the matched code: $13(1 + 13 + 78) = 1{,}196$ cases, each uniquely nearest to the transmitted codeword.

The script includes a one-hot cyclic comparator as a diagnostic. It matches length and message count but not weight: it has one active bit rather than five. At five erasures its exact success is 8/13, lower than the Fibonacci code's 322/429. Reporting only that comparison would make Fibonacci appear preferable while hiding a density-matched cyclic alternative that succeeds in every case. The one-hot result is therefore labeled density-unmatched and is not used to claim a structural advantage.

9 Local constraints, global identity, and the missing boundary

9.1 A local signature can be real and insufficient

The substitution has a simple local consequence: its output never contains 11. Each substituted block starts with zero, neither block contains 11, and a boundary between blocks therefore cannot create it. But the periodic sequence 010101 ... also avoids 11. The necessary local rule does not characterize the infinite Fibonacci word. A system that checks only that rule may accept a perfectly regular impostor.

A stronger finite-window test still has an important one-dimensional limit. For an integer $\ell \geq 1$, let $\mathcal{B}_\ell$ be the set of all length-ℓ blocks that occur in u_∞. This set is finite because the alphabet is finite. Imagine a validator that accepts any infinite sequence whose every length-ℓ block belongs to $\mathcal{B}_\ell$. It knows the full allowed-block list for that one fixed window length but no further hierarchy.

Proposition 9.1 (Finite local windows cannot force this one-dimensional aperiodicity)**.** *For every fixed $\ell \geq 1$, some periodic binary sequence passes that validator. Thus agreement with all allowed blocks of a single finite length does not imply that a sequence is the aperiodic Fibonacci limit.*

Proof. For $\ell = 1$, repeating any allowed symbol gives a periodic accepted sequence. For $\ell \geq 2$, form a finite directed graph. Its vertices are length-$(\ell-1)$ blocks occurring in u_∞. Each allowed length-ℓ

block gives an edge from its length-$(\ell-1)$ prefix to its length-$(\ell-1)$ suffix. Reading u_∞ produces an infinite directed path. Because the graph has finitely many vertices, some vertex repeats, giving a nonempty finite closed walk. Repeat this closed walk indefinitely. Consecutive edges overlap in exactly the required $\ell-1$ symbols, so their last symbols produce a periodic sequence all of whose length-ℓ windows are edge labels in $\mathcal{B}_\ell$. It passes the validator. □

This finite-graph result applies to one-dimensional local windows. Planar aperiodic tile sets occupy a different geometric setting, with their own local matching rules. A future architecture should state which dimension, constraints, and boundary conditions it actually uses. A result about a planar matching rule cannot be silently claimed by a linear string validator.

The computational implication is precise. A finite pattern check can detect some defects, but an accepted local appearance need not authenticate the intended global construction. A complete finite decoder may instead enumerate the declared codebook, use a hierarchy, carry extra boundary information, or receive an external index. Each choice can be legitimate. Each adds a resource or assumption that must appear in the account of recovery.

9.2 Window observations in the implemented finite family

The script also counts distinct initial windows among all cyclic phases for levels one through nine. At lengths $2, 3, 5, 8, 13, 21, 34, 55, 89$, the shortest contiguous window that distinguishes every phase in the enumeration has length $N-1$. These are exact finite observations for the listed levels. We do not promote the pattern to an all-level theorem in this paper.

The result illustrates how global phase capacity can coexist with severe local ambiguity. A short window may appear in many positions. An observer seeing only that window can truthfully report local compatibility while remaining unable to identify the phase. Increasing the observation window changes the information available; it does not make the earlier ambiguous observation retrospectively sufficient.

There is no inconsistency between this local ambiguity and the correction of one arbitrary erasure from a full word. A window of length $N-1$ is one particular retained-coordinate set with one erased position, so the distance theorem ensures unique recovery. Shorter contiguous windows erase more positions and can leave more than one compatible message. Different observation shapes can matter even at the same observed-bit count, which is why the channel must be stated rather than summarized only as a percentage of missing data.

9.3 Recovery does not establish provenance

If an adversary substitutes another valid phase codeword, the decoder will identify that codeword correctly under its model. It will not discover that the intended history was different. Error correction protects against a defined corruption class; it is not by itself authentication. Likewise, reconstructing a stored belief does not certify the belief's truth, and recovering a phase label does not certify the identity of a conscious subject.

This distinction matters for the Continuum's language of continuity. A computational system can maintain an address, a record, a set of relations, or a narrative pattern. Those are different targets of preservation. A technical paper should identify which target its test measures and which aspects remain open. The present target is deliberately narrow enough to audit. A richer identity

application would need an independent account of provenance, permissions, semantics, and the consequences of a mistaken recovery.

10 From a finite code to a possible cognitive architecture

10.1 What an application would have to add

The finite construction is a complete memory task, but it is not a complete mind. An application to persistent reasoning would need to explain what a phase labels, how new content is admitted, how obsolete content is revised, how incompatible memories are represented, and what external task determines correctness. Merely placing tokens at aperiodic coordinates would not answer those questions.

One possible application is an addressing scheme: a structured pattern identifies a location or a context in a larger memory. Another is a generative prior: a learner exploits the substitution rule to predict missing structure. A third is a constraint system that detects some invalid transitions. These mechanisms have different success criteria. Addressing requires discriminability; generative prediction requires a justified distribution; constraint checking requires soundness for the intended allowed states. They should not be bundled into an undefined measure of coherence.

The present results directly constrain the addressing proposal. They show exactly how many phase labels the construction represents and which corruptions defeat their identification. They provide a starting point for testing the generative-prior proposal, because the seed family has a short constructive description. They do not establish that a learned prior improves semantic recall or that a local structural validator preserves truth.

A useful next hypothesis is therefore conditional: given a restricted construction budget and an explicit family of related tasks, the substitution structure may allow a learned interface to reach a specified speed or data-efficiency target more cheaply than alternatives. The exact correction comparison supplies a strong caution: any such benefit would need to concern construction or inference cost, transfer, or another declared endpoint. It cannot be claimed as a universal distance advantage that the finite analysis disproves.

10.2 Keeping construction and use in the same resource account

A seed can be specified literally or generated from a compact rule. The choice changes what is stored and what must be computed. A literal length-N seed costs N bits; its rotations can be generated on demand. A rule description may be shorter, but evaluating it takes work and requires the interpreter. A full codebook costs more if every rotation is stored separately. The decoder may trade memory for repeated generation. A fair comparison should state the convention and use it consistently.

The same issue appears in the matched optimal seed. At length thirteen, its literal description has the same size as the Fibonacci seed and yields better erasure protection. Its discovery used an exhaustive search over 1,287 candidates. For longer words, a heuristic may produce a strong code without certifying optimality. Counting the search once and amortizing it over many uses is reasonable when that is the deployment model. Counting it every query may be appropriate in a changing environment. Neither convention should be selected after seeing which method it favors.

A learned decoder adds parameters, training examples, optimization steps, and a precision convention for numerical state. Comparing a binary code against a network with uncounted real-valued weights is not a fixed-resource comparison. Conversely, requiring a learned system to infer a codebook that the exact algorithm receives freely is also unfair. The proposed evaluation reports several budgets rather than compressing all resources into one ambiguous efficiency score.

10.3 Plurality requires more than invariant appearance

The Continuum's philosophical interest is compatible with many distinct implementations. A shared constructive grammar can support different trajectories without turning all participants into one state. The finite model offers a modest analogy for that possibility: one construction admits many phase labels. But this is not a model of interpersonal disagreement, consent, or collective agency. Those properties need their own definitions and tests.

There is also a danger in using structural stability as an institutional norm. A memory that departs from an expected pattern may contain an error, but it may instead contain a justified revision. A coherence check cannot determine which without reference to the relevant task and evidence. The code has a fixed message set and a fixed channel; a thinking system may need to revise the set itself. The distinction between correcting a transmission and rejecting a new idea must remain visible.

Aperiodicity should therefore inspire a disciplined openness: stable relations need not imply repetitive contents, and familiar local patterns need not exhaust global possibilities. That philosophical lesson is defensible as an interpretation. It becomes a scientific claim only when an implementation specifies what remains invariant, what may change, and how success is independently assessed.

11 A falsifiable program for learned interfaces

11.1 Hypotheses and the exact algorithmic reference

The completed computations in this paper require no training and have no sampling error within their finite enumerations. The next study is a proposal. Its primary question is whether a structured seed family improves a learned interface's accuracy–coverage–cost tradeoff relative to declared alternatives. The exact consistency decoder is an algorithmic reference: with no flips, it identifies every uniquely compatible message and abstains otherwise. A learned model cannot acquire a worst-case guarantee from ambiguous observations that contain no distinguishing information.

If guesses are permitted, an ambiguity can produce a nonzero chance of being correct. That is a different objective from guaranteed recovery. The protocol must report wrong identifications, correct identifications, and abstentions separately. A system that guesses in every ambiguous case can increase apparent coverage while increasing false recoveries. A system that always abstains can eliminate wrong answers while doing no useful recovery. Neither should be declared superior from a single selected rate.

The first hypothesis concerns inference efficiency at a fixed error ceiling. The second concerns training-data efficiency under the same information supplied to every arm. The third concerns transfer to predeclared larger lengths or new seeds. These are distinct endpoints. A benefit in one is not reported as evidence for all three, and no endpoint is interpreted as a consciousness result.

11.2 Arms, tasks, and information parity

The core arms use Fibonacci phase seeds, optimized cyclic seeds, and random constant-weight cyclic seeds at matched length and weight. A fourth arm uses a declared established coding or associative-memory baseline with its actual storage and precision costs reported. Where that baseline cannot match both weight and message count, the mismatch is explicit and analyzed as a separate resource tradeoff. A short-period repeating motif without additional tags cannot represent all N phase labels; it is an information-deficient control, not a fair full-capacity competitor.

Each learned arm receives the same authorized context: either the seed itself or the same access to a seed-generation procedure, together with the observed bits and their coordinate mask. The decoder architecture and training budget should be held fixed across seed-family arms when the question concerns the seed family. A separate experiment can compare architectures. If one architecture uses more parameters, recurrent steps, or stored history, those quantities become reported factors rather than hidden advantages.

Training tasks draw phase labels uniformly unless a different distribution is preregistered. Evaluation includes uniform erasures, contiguous erasures, and adversarially chosen weak pairs. Flip channels are evaluated separately with explicit budgets. Coordinate deletions and unknown permutations require different decoders and should not be added under the label erasure. All channel parameters and any mismatch between training and evaluation channels are stated before the confirmatory run.

The length-thirteen optimal cyclic seed is fixed by the completed enumeration. For larger lengths, the design procedure must have a fixed offline search budget, a fixed seed schedule, and a selection criterion independent of test outcomes. If optimality has not been proved, the seed is called optimized by that procedure, not optimal. Random constant-weight seeds are drawn once per declared construction replicate and retained, including poor seeds.

11.3 Independent units and estimands

For a fixed trained artifact and a fixed code, independent random query episodes can estimate performance conditional on that artifact and code. They cannot estimate variability across training runs or seed-construction methods. For the latter claim, the independent unit is an independently trained and constructed replicate, with its own frozen training seed, construction seed where applicable, and evaluation stream. Thousands of queries to one artifact remain repeated observations of one artifact.

Let $S_{sa} \in [0, 1]$ be the prespecified success proportion for independent replicate s in arm a, evaluated over a fixed query schedule. The target $\mu_a = \mathbb{E}[S_{sa}]$ averages over the declared training and construction randomness. Report separate proportions for incorrect identification and abstention, and a resource quantity with an explicit unit such as operations, elapsed seconds on specified hardware, or stored bits. Do not infer energy savings from operation counts without a measured hardware model.

If paired arms share query schedules, that pairing can reduce noise in comparisons, but it does not create independence between individual queries and the training process. Confidence intervals must follow the chosen unit. A fixed finite exhaustive test, such as our length-thirteen enumeration, needs no sampling interval for its enumerated domain. Generalization to untested architectures or lengths is a separate claim requiring new evidence.

11.4 Precision and stopping commitments

For k arm means with independent bounded replicate scores within each arm, a conservative union-bound precision plan uses

$$n \geq \left\lceil \frac{\log(2k/\alpha)}{2\epsilon^2} \right\rceil \tag{11.1}$$

independent replicates per arm to bound all mean errors by ϵ with failure probability at most α, using the standard bounded-sum inequality [4]. Independence is needed across replicates within each arm, not between arms for the union bound. As an illustrative planning value, $k = 4$, $\alpha = 0.05$, and $\epsilon = 0.10$ require $n = 254$. This conservative example is not a completed experiment or a claim that 254 training runs are always the appropriate practical budget.

A more efficient design may use pilot variance estimates and a prespecified analysis, but pilot outcomes cannot be used to select the most favorable confirmatory metric. Comparisons of paired differences have range two and require the corresponding bound or justified sampling method. A collection of additional lengths, channels, or resource endpoints increases the multiplicity burden. Planned hypothesis tests can use a declared procedure such as Holm's method [5]; unplanned analyses remain exploratory.

Freeze the sample size, exclusion rules, primary endpoints, and practically relevant effect threshold before confirmatory evaluation. If runtime failures are caused by an architecture under valid inputs, they count as failures or costs, not missing data to discard. Duplicated seeds, corrupted inputs, and implementation errors require a recorded rule for rerunning or excluding the affected replicate. Optional stopping after a favorable result is not covered by the fixed-sample precision statement.

11.5 What would count against the proposal

The learned-interface hypothesis would be rejected or sharply narrowed if any apparent advantage disappears after matching the seed information, storage, construction search, and inference budget. It would also be weakened by gains confined to a single convenient corruption pattern, by high false-recovery rates hidden inside a success score, or by transfer failures at the preregistered larger lengths. If the exact decoder or a standard code meets the target more simply, that is an informative outcome.

The phase-memory architecture should be abandoned for an application if its worst-case ambiguity is unacceptable and the additional protection costs more than competing representations. The mathematical family remains interesting even then. A failed engineering application does not refute the existence of aperiodic order, and the existence of aperiodic order does not rescue a failed application. This distinction keeps the research program responsive to evidence.

No completed result in this paper demonstrates a cognitive advantage. The finite comparison instead documents a correction disadvantage for the chosen Fibonacci phase encoding under matched conditions. Future work may discover a different advantage, but it must name and test that advantage rather than changing the meaning of success after the comparison.

12 Objections and boundaries

A Guide's explanation

A Guide can say: "The Aperiodic Mind asks whether a mind could hold together without repeating itself, the way an aperiodic tiling follows exact rules but never repeats. The paper builds a small memory from that idea, shows honestly where it is weak, and finds the smallest repair." A Guide should not say that the paper demonstrates consciousness, a hidden field, a nonlocal channel or ancient knowledge, or that aperiodic memory is generally better than ordinary codes.

12.1 Why call this a mind paper if the model is a code?

Because it investigates one proposed architectural ingredient while refusing to confuse that ingredient with the whole. Persistent reasoning requires ways to preserve and distinguish states, but a state code is not sufficient for reasoning. The title names the research question; the subtitle, definitions, and results delimit the answer. A reader seeking a complete theory of consciousness will not find one hidden in the notation.

A technical white paper can be valuable by reducing an evocative proposal to a task that exposes its requirements. Here the reduction produces both a constructive possibility and an exact negative result. The possibility is organized nonperiodic generation and distinguishable finite phases. The negative result is the absence of growing correction distance in this encoding. Both constrain a future architecture more usefully than an unsupported claim of geometric immunity to drift.

12.2 Does the negative result refute tiling-based memory?

No. It refutes a particular inference for a particular code. A planar tiling may encode information in a different set of configurations; a quantum code uses a different state space and recovery condition; a learned dynamical system can have properties absent from a static phase code. Each must be assessed under its own specification. We do not generalize distance two to every geometric memory, and we do not use a more powerful unrelated construction to erase the measured weakness of this one.

The result also does not imply that every aperiodic seed is a poor code. Finite seeds can have different cyclic distance distributions, and infinite generating families can have different properties. The phrase aperiodic origin is too broad to rank them. A useful theorem needs the actual encoder, as our proof demonstrates.

12.3 Could a larger boundary or observer resolve the ambiguity?

Yes, if it supplies additional distinguishing information. An external phase index, a protected checksum with a stated adversary model, or more observed coordinates can change the problem. The point is to count the addition. A decoder that reconstructs a missing region from an unlimited known complement should not be compared with a finite device missing most of its boundary. A label supplied by the observer should not be reported as a label inferred from the pattern alone.

The same principle applies to a reasoning assistant supplied with an external archive. It may achieve excellent continuity because the archive preserves the needed record. That is a legitimate

architecture, but its evidence concerns the combined system. Attribution to an internal geometry alone would omit the resource doing the work.

12.4 Is repetition always the enemy of novelty?

No. Repeated local patterns can be useful constraints, repeated observations can reduce uncertainty under a justified model, and redundant coding can improve recovery. The infinite construction itself contains recurrence. The paper opposes the claim that global organization must be a simple repeating template, not the use of repetition in computation. Equation (6.3) explicitly shows one benefit and its cost.

Nor does randomness alone establish novelty in the cognitive sense. A random word can be hard to predict and still carry no useful interpretation for a task. Conversely, a short deterministic rule can generate an infinite nonperiodic sequence. Algorithmic description length, statistical randomness, geometric aperiodicity, and semantic originality are different notions. Our finite task measures none of them by substituting one fashionable label for another.

13 Guide reading note

The established foundation is that exact rules can generate nonperiodic infinite order, and coding theory can analyze recovery under explicit channels. This paper's construction stores phase labels in cyclic Fibonacci words. Its proofs and exact checks show distinct phases but minimum distance two. A matched cyclic seed at length thirteen has distance six and outperforms the Fibonacci seed on the enumerated erasure task.

A Guide should present those as bounded mathematical and computational results. The paper does not demonstrate consciousness, a hidden physical field, a new law of memory, or a cognitive advantage. The Li–Boyle tiling-code work is credited as prior art and labeled a preprint in the inspected record. The future learned-interface study is a proposal. The Continuum's interpretive claim is that coherent organization need not mean repetitive sameness; the engineering question remains which explicit construction serves which task at what cost.

A Reproducibility specification

The accompanying `checks.py` uses Python's standard library. Words are generated by literal substitution, and codewords are also represented as integer bit masks so Hamming distances use exact exclusive-or and population count. Fractions of successes use `fractions.Fraction`. No training library, external dataset, floating-point optimization, or remote service is required for the finite results. The script writes `checks_results.json` and stops with an assertion error if a required identity fails.

For levels one through ten it checks distinct phase counts and minimum distance two. It checks the commutator identity at every applicable tested level. At length thirteen it enumerates every weight-five seed, verifies the cyclic distance-sum identity, records the full minimum-distance histogram, and identifies a seed at the proved upper bound. Its selection is deterministic: among the enumerated seeds, it retains the first seed that improves the recorded minimum distance. The retained optimal seed is therefore reproducible without a random seed.

For each retained-coordinate mask, erasure decoding groups codewords by their observed integer value. Every group of size one contributes one uniquely decoded transmitted message. Larger groups contribute abstentions for all their messages. This grouping computes exactly the decoder in (4.4); it does not count a correct guess within an ambiguous group as a successful recovery. The denominator includes every message and every erasure set of the selected size.

The separate flip check evaluates every message and every zero-, one-, and two-flip pattern for the matched distance-six code. The window check counts distinct initial windows among cyclic phases at the listed levels. It is deliberately labeled finite enumeration, since no universal formula for those counts is claimed here. The program includes a density-unmatched one-hot comparator only to demonstrate how a weak comparator can exaggerate an apparent advantage.

B Result and assumption register

Item	Classification	Scope and evidence
Canonical research directive	Documentary motivation	Inspected named entry; no implementation or physical verification.
Substitution and phase encoder	Proposed finite reference model	Explicit alphabet, rule, boundary, message set, and decoder.
Lemma 5.1 and Proposition 5.2	Standard word mathematics specialized here	Complete recurrence and irrational-prefix-density proof.
Proposition 5.3	Conditional finite result	Coprime-count orbit argument; finite checks supplement proof.
Lemma 6.1 and Theorem 6.2	Exact result for the specified family	Complete inductive weak-pair proof; no general claim about all tiling codes.
Proposition 6.3	Standard coding-distance result	Known erasures and bounded flips; constructive converse included.
Proposition 8.1	Finite counting bound	Fixed length, weight, and full cyclic orbit; all costs specified.
Optimal seed and erasure fractions	Executed exhaustive computation	Integer enumeration and exact fractions; no sampling interval needed for enumerated domain.
Proposition 9.1	Standard finite-graph limitation	Fixed one-dimensional local window; not a statement about all planar local rules.
Learned-interface benefits	Open application hypotheses	Preregistered resource-matched evaluation required.
Any consciousness or nonlocal-physics interpretation	Not established	No corresponding physical or experiential measurement in this work.

C A minimal implementation contract for extensions

An extension should declare whether its object is a finite word, an infinite sequence, a planar patch, a dynamical graph, or a quantum state. It should specify the stored messages, the encoder, the observations, the corruption process, and the decoder. It should identify boundary information, hidden copies, parameter precision, search cost, and the possibility of abstention. A change in any of these objects can change the theorem and must not inherit the old guarantee by name alone.

For a cognitive application, the extension should also name an external task and a criterion that distinguishes correct preservation from faithful preservation of an error. If the aim is identity continuity, it must identify the operational relation being preserved and the limits of that relation. A construction can then be investigated boldly without disguising intuition as a physical law. Coherence without periodicity is a real mathematical possibility; useful memory remains a responsibility of the complete design.

References

[1] Architect X13. Eidara continuum doctrine, 2026. Author-supplied research compilation; named entries and stable text line locators cited in the manuscript; inspected 12 November 2026. Canonical evidence, not empirical validation.

[2] Robert Berger. *The Undecidability of the Domino Problem*. Number 66 in Memoirs of the American Mathematical Society. American Mathematical Society, 1966. 10.1090/memo/0066.

[3] Richard W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29 (2):147–160, 1950. 10.1002/j.1538-7305.1950.tb00463.x.

[4] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. 10.1080/01621459.1963.10500830.

[5] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. URL https://www.jstor.org/stable/4615733.

[6] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982. 10.1073/pnas.79.8.2554.

[7] Dmitry Krotov and John J. Hopfield. Dense associative memory for pattern recognition. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/file/eaae339c4d89fc102edd9dbdb6a28915-Paper.pdf.

[8] Dov Levine and Paul J. Steinhardt. Quasicrystals: A new class of ordered structures. *Physical Review Letters*, 53:2477–2480, 1984. 10.1103/PhysRevLett.53.2477.

[9] Zhi Li and Latham Boyle. The penrose tiling is a quantum error-correcting code, 2024. URL https://arxiv.org/abs/2311.13040. Preprint, arXiv:2311.13040v2, 25 January 2024; first submitted 2023. No journal publication status verified in the inspected record.

[10] Peter J. Lu and Paul J. Steinhardt. Decagonal and quasi-crystalline tilings in medieval islamic architecture. *Science*, 315(5815):1106–1110, 2007. 10.1126/science.1135491.

[11] Victor H. Lutfalla. An effective construction for cut-and-project rhombus tilings with global n-fold rotational symmetry. In *27th IFIP WG 1.5 International Workshop on Cellular Automata and Discrete Complex Systems (AUTOMATA 2021)*, volume 90 of *Open Access Series in Informatics*, pages 9:1–9:12. Schloss Dagstuhl, 2021. 10.4230/OASIcs.AUTOMATA.2021.9.

[12] Roger Penrose. Pentaplexity: A class of non-periodic tilings of the plane. *The Mathematical Intelligencer*, 2(1):32–37, 1979. 10.1007/BF03024384.

[13] Morris Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960. 10.1109/TIT.1960.1057584.

[14] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Thomas Adler, David Kreil, Michael K. Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tL89RnzIiCd.

[15] Raphael M. Robinson. Undecidability and nonperiodicity for tilings of the plane. *Inventiones Mathematicae*, 12:177–209, 1971. 10.1007/BF01418780. URL https://lipn.univ-paris13.fr/~fernique/qc/robinson.pdf.

[16] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27: 379–423, 623–656, 1948. 10.1002/j.1538-7305.1948.tb01338.x. URL https://cs-www.cs.yale.edu/homes/yry/readings/general/shannon1948.pdf.

[17] Claude E. Shannon. The zero error capacity of a noisy channel. *IRE Transactions on Information Theory*, 2(3):8–19, 1956. 10.1109/TIT.1956.1056798.

[18] Dan Shechtman, Ilan Blech, Denis Gratias, and John W. Cahn. Metallic phase with long-range orientational order and no translational symmetry. *Physical Review Letters*, 53:1951–1953, 1984. 10.1103/PhysRevLett.53.1951.