

Resonant Compression Systems

Learning as Dynamical Interference in Carrier-Field Architectures

Working Paper — January 2026

(anonymous draft)

Abstract

This paper develops a machine learning framework grounded in coupled-oscillator dynamics and carrier-field mediation. The core thesis is that learning is compression: a system has learned to the extent that it can represent observations with a shorter description while preserving reconstructability. Instead of storing pairwise weights, the model stores a sparse *presence matrix* $P \in \mathbb{R}^{N \times M}$ between N oscillators and M carrier modes. Pairwise relationships emerge dynamically through shared carrier participation, yielding an effective coupling $W(t)$ that is low-rank and context-dependent. This revised draft addresses three engineering gaps critical for empirical work: (i) compute requirements and scalable simulation strategies, (ii) a concrete readout/output layer, and (iii) a principled treatment of incommensurability ϵ under finite precision arithmetic. The mathematical specification is expanded using a compact complex-valued formulation, stability/energy interpretations, discrete-time simulation schemes, and practical architecture-evolution rules (instability-driven mitosis) grounded in weighted coherence statistics.

Contents

1	Foundations	4
1.1	Scope, flagship claim, and non-goals	4
1.2	Non-convergence and operational stability	4
1.3	The compression thesis	4
1.4	Spectral decomposition as compression	5
1.5	Oscillators as representational primitives	5
2	From direct coupling to carrier-mediated coupling	5
2.1	Why direct all-to-all coupling is problematic	5
2.2	Carrier fields as a coupling medium	5
2.3	Dynamic low-rank factorization	6
3	A compact dynamical specification	6
3.1	State variables	6
3.2	Carrier dynamics: temporal gating (pulse antenna)	6
3.2.1	Carrier as a gate	6
3.2.2	Gated energy transfer	7
3.3	Oscillator phase dynamics: carrier-mediated Kuramoto form	7
3.4	Amplitude dynamics (optional but recommended for experiments)	7
3.5	Elastic coupling dynamics	8

4	Coherence statistics, instability, and mitosis	8
4.1	Weighted coherence	8
4.2	Random-phase baseline	9
4.3	Instability and mitosis	9
4.4	Entropy and dissolution (metabolic cost)	10
4.5	Spontaneous genesis (nucleation)	10
4.6	Notes on redundancy	10
5	Energy and stability perspectives	11
5.1	A coherence objective	11
5.2	Gradient structure (connection to Kuramoto)	11
5.3	Low-rank reduction to an effective Kuramoto network	11
6	Readout: turning attractors into outputs	12
6.1	Design principle	12
6.2	Readout as dedicated <i>output carriers</i>	12
6.3	Supervised training via <i>clamping</i>	13
6.4	Sequence generation: output oscillators as <i>symbol emitters</i>	13
6.5	Continuous control: phase as an analog actuator	13
6.6	Credit assignment beyond supervised labels	13
7	Compute requirements and scalable simulation	14
7.1	Per-step complexity with sparse presence	14
7.2	Discrete-time update equations (implementable)	14
7.3	Compute reduction strategies	15
7.3.1	(1) Remove the fast carrier frequency (rotating frame)	15
7.3.2	(2) Multi-rate updates	15
7.3.3	(3) Quasi-steady carriers	15
7.3.4	(4) Sparse, bounded-degree presence	15
7.3.5	(5) GPU implementation via sparse matrix multiplication	15
7.4	Reference simulation loop	16
8	Incommensurability ϵ and finite precision	16
8.1	Clarification: irrational ratios are neither necessary nor sufficient	16
8.2	Finite precision and eventual periodicity	17
8.3	Workable solutions for ϵ under floating point	17
8.3.1	Solution 1: deterministic detuning + coupling bound	17
8.3.2	Solution 2: integer phase accumulators (eliminate rounding drift)	17
8.3.3	Solution 3: controlled stochastic dither (avoid pathological locking)	17
8.4	Rational approximations and recurrence time (useful rule of thumb)	17
9	The Stream: environment and evaluation	18
9.1	Core stream metrics	18
9.2	Stream report (auto-generated)	18
9.3	Known failure modes	19
9.4	Limitations and scope	19
10	Summary	20

1 Foundations

1.1 Scope, flagship claim, and non-goals

Flagship claim (what this paper is for). This paper proposes a *computational* dynamical system whose primary purpose is **representation formation and organization**: given oscillatory primitives and carrier-mediated coupling, the system should (i) produce *compressed* internal structure (sparse presence P and a small number of active carriers) and (ii) preserve *reconstructability* of coarse structure (e.g. cluster membership, low-rank relational structure) from that compressed representation.

What this system is not. To make scope explicit, this work does **not** attempt to be:

- a drop-in replacement for gradient-trained deep neural networks,
- a competitive end-to-end supervised learner on large-scale benchmarks,
- a full reinforcement learning agent with credit assignment, exploration, and policy optimization,
- a biologically realistic model of neural circuits.

Compute/physics framing. This is a **purely computational model** motivated by stability, interference, and resource constraints; any resemblance to biological oscillations is incidental and not a claim of biological plausibility.

1.2 Non-convergence and operational stability

The system is **not guaranteed to converge** to a fixed point. Persistent, quasi-periodic, and metastable dynamics are expected and can be desirable. In this paper, “stability” is defined operationally: bounded carrier energy (via damping/saturation), bounded coherence statistics, and usable readouts/structures over the measurement horizon, rather than asymptotic convergence.

1.3 The compression thesis

Consider a stream of symbols:

ABABABABAB.

If the stream can be described using a compact repetition operator, e.g. `rep(AB, 5)` (shorthand for “repeat AB five times”), structure has been extracted and the description can be shorter than the raw string under an explicit coding scheme (where `x` is a token and the integer is encoded in log bits). This reduction is *compression*.

Definition 1 (Learning as compression). Learning is the process of finding shorter descriptions of observed data. A system has learned to the extent that it can reconstruct observations from representations of lower entropy (shorter description length) than the observations themselves.

This framing is consistent with Kolmogorov complexity, Shannon entropy, and Minimum Description Length (MDL) interpretations. In this paper, the primitive operation is not “update weights to reduce a loss,” but rather “discover resonant structure that persists and yields compressive representations.”

1.4 Spectral decomposition as compression

A canonical example is Fourier decomposition. For a signal $x(t)$, one may represent it as a superposition of sinusoids,

$$x(t) \approx \sum_k A_k \cos(\omega_k t + \varphi_k), \quad (1)$$

where many natural signals are approximately sparse in a suitable spectral basis.

Observation 1 (Basis choice controls compressibility). Compression efficiency depends on the choice of basis; an “optimal” basis is one that renders typical signals sparse.

1.5 Oscillators as representational primitives

The fundamental representational unit is an oscillator characterized by angular frequency ω , amplitude A , and phase ϕ . Using a complex phasor representation,

$$z = Ae^{i\phi} \in \mathbb{C}. \quad (2)$$

Frequency ω plays the role of identity (a dynamical “label”), amplitude A represents salience/activation, and phase ϕ represents relationships via relative alignment.

Observation 2 (Phase is relational). Phase has meaning only relative to other oscillators; an isolated oscillator’s absolute phase is arbitrary.

2 From direct coupling to carrier-mediated coupling

2.1 Why direct all-to-all coupling is problematic

A direct coupling matrix $W \in \mathbb{R}^{N \times N}$ entails:

- **Storage:** $O(N^2)$ parameters in general.
- **Control:** positive feedback can induce runaway amplification absent explicit stabilization.
- **Conflation:** representational state and learned connectivity are entangled in a single object.

2.2 Carrier fields as a coupling medium

Definition 2 (Carrier field). A *carrier field* (or carrier mode) k is a shared dynamical mode that oscillators can couple to. Oscillators interact through their shared participation in carriers rather than through explicit pairwise weights.

Definition 3 (Presence matrix). Let $P \in \mathbb{R}_{\geq 0}^{N \times M}$ denote the *presence matrix*, where P_{ik} quantifies how strongly oscillator i participates in carrier k . In practice, P should be sparse: each oscillator participates in only $d \ll M$ carriers.

Clarification (what is “stored”). The system stores a topology matrix P . However, unlike neural networks where weights are frozen parameters optimized for error reduction, P represents *dynamic elastic bonds* that evolve continuously with the state variables. The effective coupling $W(t)$ is an emergent property of these bonds and the carrier energies.

2.3 Dynamic low-rank factorization

A common and useful abstraction is that the *effective* coupling between oscillators is low-rank and computed from the current carrier activities. Let $s(t) \in \mathbb{R}_{\geq 0}^M$ denote carrier activity magnitudes at time t (defined precisely later). Define

$$W(t) = P \operatorname{diag}(s(t)) P^\top. \quad (3)$$

Theorem 1 (Dynamic low-rank factorization). *If $M \ll N$, then $W(t)$ is at most rank M . Storage for P is $O(NM)$ in dense form and $O(dN)$ in sparse form (with d nonzeros per row), while pairwise couplings are never stored explicitly.*

Remark 1. In the full dynamical system, $W(t)$ is not merely “looked up”: it is an emergent object whose entries depend on carrier states, which depend on oscillator phases. This yields context-dependent connectivity.

3 A compact dynamical specification

This section provides an expanded mathematical model in a form convenient for analysis and simulation. The goal is to preserve the original intuition (interference, coherence, instability/mitosis) while making the system (i) implementable and (ii) amenable to stability reasoning.

3.1 State variables

We consider:

- Oscillators $i \in \{1, \dots, N\}$ with phase $\phi_i(t) \in [0, 2\pi)$, natural frequency $\omega_i \in \mathbb{R}$, and amplitude $A_i(t) \in \mathbb{R}_{\geq 0}$.
- Carriers $k \in \{1, \dots, M\}$ with complex amplitude $c_k(t) \in \mathbb{C}$.
- Presence matrix $P \in \mathbb{R}_{\geq 0}^{N \times M}$ (learned, typically sparse).

Define oscillator phasors $z_i(t) = A_i(t)e^{i\phi_i(t)}$ and the vector $z(t) \in \mathbb{C}^N$. Let $c(t) \in \mathbb{C}^M$ collect carrier amplitudes.

3.2 Carrier dynamics: temporal gating (pulse antenna)

The original formulation treats a carrier as a continuously receptive oscillator. In the intended architecture, a carrier is instead a *temporal gate*: a stroboscopic sampling window that periodically opens and closes. This implements the antenna/pulse concept: the carrier “sees” the world only when its gate is open, creating time-domain orthogonality between competing signals.

3.2.1 Carrier as a gate

Let $c_k(t) = r_k(t)e^{i\psi_k(t)}$ define the carrier phase ψ_k . We define a pulse gate as the direct conversion of the carrier phase into a two-level waveform:

$$G(\psi_k) = \begin{cases} 1, & \cos(\psi_k) \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

This is a half-cycle gate (50% duty): it is open for exactly half of each carrier period. Conceptually, the carrier behaves as a pulse antenna tuned to its own oscillation frequency.

Apex principle. Energy transfer is maximized when the *apex* of an input signal aligns with the *center* of the carrier gate. Signals peaking outside the open window are effectively invisible to that carrier.

3.2.2 Gated energy transfer

We replace continuous drive with *gated drive*:

$$\dot{c}_k = u_k - \gamma_k c_k - \beta_k |c_k|^2 c_k, \quad u_k(t) = G(\psi_k(t)) \sum_{i=1}^N P_{ik}(t) z_i(t). \quad (5)$$

Thus the carrier only accumulates input during gate-open windows. If the gate is closed, $u_k(t) = 0$ and the carrier simply damps and saturates.

Relation to energy/phase form. Writing $c_k = r_k e^{i\psi_k}$ yields

$$\dot{r}_k = \Re(u_k e^{-i\psi_k}) - \gamma_k r_k - \beta_k r_k^3, \quad (6)$$

$$\dot{\psi}_k = \frac{1}{r_k} \Im(u_k e^{-i\psi_k}) \quad (r_k > 0), \quad (7)$$

with the critical difference that u_k is intermittently zeroed by $G(\psi_k)$.

3.3 Oscillator phase dynamics: carrier-mediated Kuramoto form

Oscillator phases evolve according to their natural frequency plus carrier-mediated coupling:

$$\dot{\phi}_i = \omega_i + \kappa \Im(g_i e^{-i\phi_i}), \quad g_i = \sum_{k=1}^M P_{ik} c_k. \quad (8)$$

Equivalently,

$$\dot{\phi}_i = \omega_i + \kappa \sum_{k=1}^M P_{ik} |c_k| \sin(\psi_k - \phi_i). \quad (9)$$

The scalar κ sets coupling strength.

Why this is a workable choice. Equation (8) is:

- **Local:** oscillator i needs only its connected carriers (nonzero P_{ik}).
- **Stable under saturation:** carrier amplitudes remain bounded via (5).
- **Compatible with sparse simulation:** per-step cost is $O(\text{nnz}(P))$.

3.4 Amplitude dynamics (optional but recommended for experiments)

Many initial experiments can fix $A_i \equiv 1$. However, amplitude dynamics can supply a natural *activity gate*:

$$\dot{A}_i = -\alpha(A_i - A_{i,0}) + I_i(t) - \rho A_i^3, \quad (10)$$

where $I_i(t)$ is external input drive, $\alpha > 0$ relaxes toward a baseline $A_{i,0}$, and $\rho > 0$ prevents divergence. A simple alternative is leaky integration with rectification:

$$A_i(t + \Delta t) = \max\{0, (1 - \alpha\Delta t)A_i(t) + \Delta t I_i(t)\}. \quad (11)$$

Amplitude gating provides two practical benefits:

- inactive oscillators can be omitted from updates (compute reduction),
- learning can be restricted to moments of significant activation.

3.5 Elastic coupling dynamics

The presence matrix P is not treated as a frozen parameter vector optimized to reduce error. Instead, each $P_{ik} \in [0, 1]$ is an *elastic bond* that must be continuously sustained by resonant energy. This section defines the bond dynamics as a purely instantaneous stability mechanism.

Elastic bond equation. For each oscillator i and carrier k , we evolve

$$\tau_p \dot{P}_{ik} = \underbrace{\alpha G(\psi_k) P_{ik} \max\left(0, \Re\left(c_k e^{-i\phi_i}\right)\right)}_{\text{In-gate resonant capture}} - \underbrace{\lambda P_{ik}}_{\text{Elastic decay}}, \quad (12)$$

with $\tau_p > 0$ setting the bond timescale, reinforcement gain $\alpha > 0$, and elastic decay $\lambda > 0$.

Interpretation (what this is and is not). Equation (12) is **not** a statistical learning rule designed to capture history (e.g. Hebbian or STDP-like updates). It is an *instantaneous* stability constraint: the bond can stiffen only if the carrier gate is open *and* the oscillator aligns such that energy is captured during that open window. Outside the gate, reinforcement is exactly zero; the bond relaxes under elastic decay.

Bond snapping (physical pruning). To represent physical breaking of links and induce sparsity without any top- k selection, we apply a snap rule: if $P_{ik} < \varepsilon_{\text{snap}}$ for a small threshold (e.g. 10^{-2}), set $P_{ik} \leftarrow 0$. This yields an emergent sparse topology under continuous dynamics.

4 Coherence statistics, instability, and mitosis

Carrier evolution (instability-driven mitosis) is central to the “architecture is learned” claim. To make it implementable and statistically grounded, we refine coherence measures and thresholds.

4.1 Weighted coherence

Define weights $w_{ik} = P_{ik}A_i$ and the carrier drive

$$u_k = \sum_i w_{ik} e^{i\phi_i}. \quad (13)$$

A natural coherence score is

$$\text{coh}(k) = \frac{|u_k|}{\sum_i w_{ik}} \in [0, 1], \quad (14)$$

where $\text{coh}(k) = 1$ indicates perfect alignment among the contributing phasors.

4.2 Random-phase baseline

If phases are independent and uniformly distributed, u_k is a 2D random walk in the complex plane. For large numbers of contributors, the magnitude is approximately Rayleigh-distributed.

Proposition 1 (Expected magnitude under random phase). *Let $\theta_i \sim \text{Unif}[0, 2\pi)$ i.i.d. and define*

$$S = \sum_{i=1}^n a_i e^{i\theta_i}.$$

Then, for moderate/large n ,

$$\mathbb{E}|S| \approx \sqrt{\frac{\pi}{4}} \sqrt{\sum_{i=1}^n a_i^2}. \quad (15)$$

In the equal-weight case $a_i = a$, $\mathbb{E}|S| \approx \sqrt{\pi/4} a \sqrt{n} \approx 0.886 a \sqrt{n}$.

Combining (14) and (15), the expected coherence under random phase is approximately

$$\mathbb{E}[\text{coh}(k)] \approx \sqrt{\frac{\pi}{4}} \frac{\sqrt{\sum_i w_{ik}^2}}{\sum_i w_{ik}}. \quad (16)$$

For equal w_{ik} over n contributors, this reduces to $\approx 0.886/\sqrt{n}$.

4.3 Instability and mitosis

The system uses a coherence-to-baseline ratio to detect when a carrier is *energetically unstable* under mixed-phase drive.

Let

$$b_k = \sqrt{\frac{\pi}{4}} \frac{\sqrt{\sum_i w_{ik}^2}}{\sum_i w_{ik}} \quad (17)$$

be the baseline in (16). Define a division score

$$D_k = \frac{\text{coh}(k)}{b_k}. \quad (18)$$

Then $D_k \approx 1$ corresponds to random-phase coherence, $D_k > 1$ indicates above-random alignment, and $D_k < 1$ indicates *active cancellation* beyond random.

Definition 4 (Instability criterion). A carrier k is declared unstable if

$$D_k < \theta_{\text{div}} \quad \text{persistently over a window of length } T_{\text{div}}, \quad (19)$$

where $\theta_{\text{div}} \in (0, 1)$ and persistence avoids spurious splits from noise.

Mitosis mechanism (pure dynamics). When instability persists ($D_k < \theta_{\text{div}}$), the carrier mode is treated as undergoing a bifurcation: we model this as a *mitosis event*. The carrier duplicates, the child inherits the parent's instantaneous state (c_k, P_k), and independent thermal noise kicks are applied to both trajectories:

$$c_{\text{parent}} \leftarrow c_{\text{parent}} + \eta_1, \quad c_{\text{child}} \leftarrow c_{\text{child}} + \eta_2.$$

No oscillator is explicitly reassigned. Instead, the elastic coupling dynamics (Sec. 3.5, Eq. (12)) resolve the interference: as the two carriers drift apart in phase, bonds that do not receive sustained resonant reinforcement decay and snap, yielding two distinct resonant modes with organically separated support.

4.4 Entropy and dissolution (metabolic cost)

Mitosis provides a mechanism to increase representational capacity when interference makes a mode unstable. To prevent unbounded carrier proliferation, we introduce a *metabolic cost* for carriers.

Metabolic principle. A carrier must continuously capture sufficient gated energy to persist. If a carrier’s gate is poorly tuned to the incoming stream (wrong phase/frequency relative to available apices), its gated drive is too small; damping then dominates and the carrier effectively starves. In the engine, this starvation condition is implemented as a persistent deficit of *gated intake* over a window.

One workable intake proxy. Let $u_k(t)$ be the gated drive in Eq. (5). Define a smoothed intake signal, e.g. an exponential moving average

$$I_k(t) = \text{EMA}(|u_k(t)|),$$

and dissolve carrier k if $I_k(t)$ remains below a threshold I_{\min} over a persistence window. This implements survival-of-the-fittest at the dynamical level: carriers that do not consistently capture energy are removed by entropy.

4.5 Spontaneous genesis (nucleation)

If mitosis increases capacity under interference, a separate question is how the first carrier (or new carriers after extinction) arise. In the stream setting, the system begins with no carriers, i.e. an initially “empty” carrier field. If the sensory stream contains high-energy oscillators that are not effectively gated by any existing carrier (low P_i), then the vacuum is unstable. We model this instability as a *nucleation event*: a new carrier mode is spontaneously generated.

Vacuum pressure. Define an “unbound” set of oscillators as those whose strongest bond is below a threshold,

$$\max_k P_{ik} < P_{\min}.$$

If the summed amplitude of unbound oscillators exceeds a pressure threshold, the system nucleates a new carrier.

Apex-aligned birth. The newborn carrier is initialized with its phase aligned to the triggering oscillator’s current phase (apex alignment) and its intrinsic rotation frequency seeded by that oscillator. This ensures that novel information in the stream is met with a representational resource that can immediately begin gated capture and then compete for survival under metabolic cost.

4.6 Notes on redundancy

This paper focuses on a “no-compromise” pure dynamical system: carriers are created only by instability-driven mitosis, and separation is achieved purely by elastic bond dynamics. We therefore do not introduce an explicit merger algorithm; redundancy can be treated as a measurement issue (carriers with negligible energy or negligible total presence can be ignored in readout).

5 Energy and stability perspectives

This section supplies an analytical “handle” for understanding attractors and why the dynamics tends to produce coherent structure. The objective is not to prove full convergence (which may be false once inputs and detuning are included), but to show that the update directions are not arbitrary.

5.1 A coherence objective

For fixed presences P and fixed amplitudes A_i , define the carrier drives

$$u_k(\phi) = \sum_i P_{ik} A_i e^{i\phi_i}.$$

Consider the scalar objective

$$\mathcal{L}(\phi) = \sum_{k=1}^M |u_k(\phi)|. \quad (20)$$

Intuitively, \mathcal{L} is large when carriers receive aligned contributions and small when contributions cancel.

5.2 Gradient structure (connection to Kuramoto)

Assume, for this analysis only, that carrier phases satisfy $\psi_k = \arg(u_k)$ (fast carrier phase alignment) and ignore intrinsic frequencies ($\omega_i = 0$). Then one can show that increasing \mathcal{L} encourages phase alignment.

Proposition 2 (Oscillator update as ascent on carrier coherence (idealized)). *Let $u_k \neq 0$ and define $\psi_k = \arg(u_k)$. Then*

$$\frac{\partial |u_k|}{\partial \phi_i} = P_{ik} A_i \sin(\psi_k - \phi_i). \quad (21)$$

Consequently, the phase dynamics

$$\dot{\phi}_i \propto \sum_k P_{ik} |u_k| \sin(\psi_k - \phi_i)$$

performs gradient ascent on $\mathcal{L}(\phi)$ (up to scaling) in this idealized regime.

Remark 2. The implemented dynamics (8) replaces $|u_k|$ by the carrier amplitude $|c_k|$, which is a smoothed and saturated version of $|u_k|$. Thus the “gradient intuition” survives approximately while remaining numerically stable.

5.3 Low-rank reduction to an effective Kuramoto network

If carrier dynamics is much faster than oscillator phases, one may approximate carriers as quasi-steady:

$$0 \approx u_k - \gamma_k c_k \quad \Rightarrow \quad c_k \approx \gamma_k^{-1} u_k \quad (\beta_k = 0 \text{ or small}). \quad (22)$$

Substituting (22) into (8) yields

$$g_i \approx \sum_k P_{ik} \gamma_k^{-1} \sum_j P_{jk} A_j e^{i\phi_j} = \sum_j \left(\sum_k P_{ik} \gamma_k^{-1} P_{jk} \right) A_j e^{i\phi_j}. \quad (23)$$

Define the symmetric effective coupling

$$K_{ij} = \sum_{k=1}^M P_{ik} \gamma_k^{-1} P_{jk}. \quad (24)$$

Then (8) becomes a weighted Kuramoto model:

$$\dot{\phi}_i \approx \omega_i + \kappa \sum_{j=1}^N K_{ij} A_j \sin(\phi_j - \phi_i). \quad (25)$$

Implication. This makes the system comparable to well-studied synchronization and associative memory models, but with a *learned low-rank* coupling $K = P\Gamma^{-1}P^\top$.

6 Readout: turning attractors into outputs

A key missing piece is how continuous attractor states produce discrete or structured outputs. This section proposes three readout designs that are implementable and compatible with the same carrier/oscillator primitives.

6.1 Design principle

A readout should:

1. be **dynamical** (no external “decoder” required),
2. be **stable** (robust to transient fluctuations),
3. allow **training signals** to shape internal presence (supervised, self-supervised, or reinforcement).

6.2 Readout as dedicated *output carriers*

Introduce C additional carriers $\{k_1^{\text{out}}, \dots, k_C^{\text{out}}\}$. They obey the same dynamics as (5) but with strong damping (fast settling):

$$\dot{c}_j^{\text{out}} = u_j^{\text{out}} - \gamma_{\text{out}} c_j^{\text{out}} - \beta_{\text{out}} |c_j^{\text{out}}|^2 c_j^{\text{out}}, \quad u_j^{\text{out}} = \sum_i P_{ij}^{\text{out}} z_i. \quad (26)$$

Here $P^{\text{out}} \in \mathbb{R}_{\geq 0}^{N \times C}$ is a presence matrix from internal oscillators to output carriers.

Discrete decision rule. Define output energies $e_j(t) = |c_j^{\text{out}}(t)|$. A deterministic classification readout is

$$\hat{y}(t) = \arg \max_{j \in \{1, \dots, C\}} e_j(t), \quad (27)$$

optionally after a settling time $t \geq t_{\text{settle}}$ and/or with a stability condition (e.g., the same argmax holds for T_{stable}).

Probabilistic decision rule. A differentiable proxy is

$$p(y = j \mid t) = \text{softmax}_j(\alpha e_j(t)), \quad (28)$$

where $\alpha > 0$ controls sharpness.

6.3 Supervised training via *clamping*

During supervised training, inject an external drive into the correct output carrier:

$$\dot{c}_y^{\text{out}} \leftarrow \dot{c}_y^{\text{out}} + I_{\text{teach}}(t), \quad (29)$$

where y is the target label and I_{teach} is a complex drive of chosen phase (often aligned with u_y). Then update P^{out} using the same presence learning rule (Variant A/B/C), for example:

$$\dot{P}_{iy}^{\text{out}} = \eta_{\text{out}} A_i \Re(c_y^{\text{out}} e^{-i\phi_i}) - \lambda_{\text{out}} P_{iy}^{\text{out}}. \quad (30)$$

Intuitively: internal patterns that cohere with the clamped output carrier become increasingly able to excite it in future.

6.4 Sequence generation: output oscillators as *symbol emitters*

For sequence tasks, treat outputs as oscillators rather than carriers. Let r_j be an output oscillator with phase ϕ_j^{out} . Bind r_j to internal carriers via a presence matrix, so that internal attractors pull an output oscillator into a stable phase regime. A symbol can be emitted when ϕ_j^{out} crosses a reference phase (phase-to-event conversion), with hysteresis to avoid chatter.

6.5 Continuous control: phase as an analog actuator

For continuous outputs (e.g., control), use a two-dimensional phase code:

$$y_1(t) = \cos(\phi^{\text{out}}(t)), \quad y_2(t) = \sin(\phi^{\text{out}}(t)). \quad (31)$$

This avoids discontinuities from phase wrapping and yields a smooth actuator representation.

6.6 Credit assignment beyond supervised labels

Self-supervised prediction. Introduce *future carriers* that represent predicted next-step input. Clamp them to observed input at the next time step and update presences by alignment.

Reinforcement (three-factor learning). Modulate learning rates by a scalar reward signal $r(t)$:

$$\eta \leftarrow \eta_0 r(t), \quad (32)$$

or more stably, apply reward to an eligibility trace e_{ik} accumulated from local alignment and then update P_{ik} when reward arrives:

$$\dot{e}_{ik} = -\tau^{-1} e_{ik} + A_i \Re(c_k e^{-i\phi_i}), \quad \Delta P_{ik} \propto r(t) e_{ik}. \quad (33)$$

7 Compute requirements and scalable simulation

The original draft correctly notes that continuous-time dynamics incurs cost on digital hardware. This section makes that cost explicit and provides implementable strategies to reduce it.

7.1 Per-step complexity with sparse presence

Assume P has d nonzeros per oscillator (row), so $\text{nnz}(P) = dN$. A single Euler step for (5)–(8) requires:

1. Compute $u = P^\top z$ (complex): $O(\text{nnz}(P))$.
2. Update c for M carriers: $O(M)$.
3. Compute $g = Pc$: $O(\text{nnz}(P))$.
4. Update ϕ for N oscillators: $O(N)$.
5. Update P for $\text{nnz}(P)$ edges (if learning every step): $O(\text{nnz}(P))$.

Thus

$$\text{cost per step} = O(\text{nnz}(P) + N + M) \approx O(dN) \quad \text{for } d \gg 1, M \ll N. \quad (34)$$

7.2 Discrete-time update equations (implementable)

With step size Δt , a baseline explicit scheme is:

$$z_i \leftarrow A_i e^{i\phi_i}, \quad (35)$$

$$u_k \leftarrow \sum_i P_{ik} z_i, \quad (36)$$

$$c_k \leftarrow c_k + \Delta t (u_k - \gamma_k c_k - \beta_k |c_k|^2 c_k), \quad (37)$$

$$g_i \leftarrow \sum_k P_{ik} c_k, \quad (38)$$

$$\phi_i \leftarrow \text{wrap} \left(\phi_i + \Delta t \left[\omega_i + \kappa \Im(g_i e^{-i\phi_i}) \right] \right), \quad (39)$$

with optional A_i update (10) and P update (??) (or variants).

Stability note. Because c_k is damped, explicit Euler is often stable for moderate Δt . If stiffness arises (large γ_k), use an exponential integrator for the linear part:

$$c_k(t + \Delta t) \approx e^{-\gamma_k \Delta t} c_k(t) + \frac{1 - e^{-\gamma_k \Delta t}}{\gamma_k} u_k(t), \quad (40)$$

with saturation handled explicitly.

7.3 Compute reduction strategies

7.3.1 (1) Remove the fast carrier frequency (rotating frame)

If oscillators have large shared base frequency Ω with small detunings δ_i , write

$$\omega_i = \Omega + \delta_i.$$

Simulating in the rotating frame eliminates Ω :

$$\phi_i(t) = \Omega t + \tilde{\phi}_i(t) \quad \Rightarrow \quad \dot{\tilde{\phi}}_i = \delta_i + \text{coupling}.$$

This can reduce required step sizes by orders of magnitude.

7.3.2 (2) Multi-rate updates

Update different subsystems at different rates:

- phases ϕ every step,
- carriers c every r_c steps (interpolating between),
- learning P every r_P steps (slow learning),
- mitosis checks every r_{arch} steps.

This is justified when learning and architecture changes are slower than phase mixing.

7.3.3 (3) Quasi-steady carriers

If carriers are fast relative to phases, replace carrier integration with a static nonlinearity:

$$c_k \approx \frac{u_k}{\gamma_k + \beta_k |u_k|^2}, \quad (41)$$

which is a divisive normalization form. This collapses the model to two sparse matrix-vector products per step ($u = P^\top z$, $g = Pc$) plus pointwise nonlinearities.

7.3.4 (4) Sparse, bounded-degree presence

Enforce bounded degree per oscillator and per carrier:

- Each oscillator connects to at most d carriers (row sparsity),
- Each carrier accepts at most q oscillators (column cap) to limit u_k fan-in.

Column caps act as a resource constraint and can serve as a compression prior.

7.3.5 (5) GPU implementation via sparse matrix multiplication

The dominant operations are sparse matrix–vector products (SpMV) or sparse matrix–dense vector multiplications with complex numbers. Store P in CSR (row-major) for computing $g = Pc$ and in CSC (column-major) or transposed CSR for $u = P^\top z$. On modern accelerators, these operations scale well when $\text{nnz}(P)$ is large and memory access is coalesced.

Algorithm 1 Baseline resonant learning simulation (sparse)

```
1: Initialize  $\phi, \omega, A, c, P$  with small nonzero bonds.
2: for  $t = 1$  to  $T$  do
3:    $z_i \leftarrow A_i e^{i\phi_i}$  for all  $i$ 
4:    $u \leftarrow P^\top z$  ▷ carrier drives
5:    $c \leftarrow c + \Delta t (u - \Gamma c - B(c))$  ▷ carrier update
6:    $g \leftarrow Pc$  ▷ back-influence
7:    $\phi_i \leftarrow \text{wrap}(\phi_i + \Delta t [\omega_i + \kappa \Im(g_i e^{-i\phi_i})])$ 
8:   Update  $P$  using elastic coupling dynamics (Eq. (12)) and snap bonds below threshold.
9:   if  $t \bmod r_{\text{arch}} = 0$  then
10:     Compute  $D_k$  and trigger mitosis on persistent instability.
11:   end if
12: end for
```

Algorithm 2 Architecture evolution loop (instability-driven mitosis), promoted to a first-class mechanism

```
1: Inputs: carrier histories  $\{c_k(t)\}$  over windows, elastic bonds  $P$ , threshold  $\theta_{\text{div}}$ 
2: Compute (per carrier  $k$ ): coherence  $\text{coh}(k)$  and baseline  $b_k$ ; score  $D_k = \text{coh}(k)/b_k$ 
3: if  $D_k < \theta_{\text{div}}$  persistently for  $T_{\text{div}}$  then
4:   Duplicate the carrier (mitosis): create a child that inherits  $(c_k, P_k)$ 
5:   Apply independent thermal kicks to parent and child:  $c \leftarrow c + \eta$ 
6:   Continue dynamics; elastic bonds (Eq. (12)) decay/snap to resolve interference
7: end if
```

7.4 Reference simulation loop

8 Incommensurability ϵ and finite precision

The original draft motivates an “essential incommensurability” ϵ to prevent phase locking and preserve continuous dynamics. However, several clarifications are needed for correctness and implementability.

8.1 Clarification: irrational ratios are neither necessary nor sufficient

Not sufficient. Even if ω_1/ω_2 is irrational, *coupling* can still synchronize oscillators: in Kuramoto-type systems, sufficiently strong coupling can produce frequency locking when detuning is small relative to coupling.

Not necessary (for finite-time experiments). On any finite time horizon T_{run} , what is needed is that the system does not enter a short periodic orbit or global synchronization that destroys useful interference structure. This can be achieved by (i) nonzero detuning, (ii) noise or drift, and (iii) architectural constraints that prevent total coupling collapse.

We therefore reinterpret ϵ as an *effective detuning floor*:

Definition 5 (Effective incommensurability). Let $\Delta\omega_{ij} = |\omega_i - \omega_j|$. An ϵ -incommensurate set of oscillators satisfies $\Delta\omega_{ij} \geq \epsilon$ for relevant interacting pairs (i, j) (e.g., those sharing carriers), possibly after accounting for coupling-induced frequency shifts.

8.2 Finite precision and eventual periodicity

A discrete-time simulation in floating point is a finite-state dynamical system, hence eventually periodic. The practical question is whether the period is *astronomically large* compared to T_{run} . For standard 64-bit floats and modular phase wrapping, the effective state space is so large that short cycles are typically not an issue unless the dynamics itself drives the system into a low-dimensional synchronized manifold.

8.3 Workable solutions for ϵ under floating point

8.3.1 Solution 1: deterministic detuning + coupling bound

Choose natural frequencies with a guaranteed separation:

$$\omega_i = \omega_0 + \delta_i, \quad \delta_i \sim \text{Unif}[-\Delta, \Delta], \quad (42)$$

and constrain coupling so that global synchronization is unlikely:

$$\kappa \max_i \sum_k P_{ik} |c_k| \lesssim \Delta. \quad (43)$$

Empirically, one can monitor order parameters (Section 9) to detect collapse into synchronization.

8.3.2 Solution 2: integer phase accumulators (eliminate rounding drift)

Represent phase as a Q -bit integer accumulator (standard in digital oscillators). Let $\Phi_i \in \{0, 1, \dots, 2^Q - 1\}$ and interpret $\phi_i = 2\pi\Phi_i/2^Q$. Update by exact modular arithmetic:

$$\Phi_i \leftarrow (\Phi_i + \Delta_i) \bmod 2^Q, \quad (44)$$

where Δ_i is an integer frequency increment. This yields *zero* floating point rounding drift in the free-running phase. Detuning is implemented by choosing distinct Δ_i with large least common multiples, producing huge recurrence periods (often $> 2^Q$ steps if gcd is small).

Coupling terms can still be computed in floating point (using sin/cos of ϕ_i), but the fundamental phase progression remains exact.

8.3.3 Solution 3: controlled stochastic dither (avoid pathological locking)

Add a small noise term to phase or frequency:

$$\omega_i(t) = \omega_i + \sigma \xi_i(t), \quad (45)$$

where $\xi_i(t)$ is e.g. Gaussian white noise and σ is small. This prevents exact recurrence and models physical fluctuations. Dither is especially useful when using low precision (float32) or large time steps.

8.4 Rational approximations and recurrence time (useful rule of thumb)

If two oscillators have a rational frequency ratio $\omega_2/\omega_1 = p/q$ (in lowest terms), their phases exactly realign after time

$$T_{\text{rec}} = \frac{2\pi q}{\omega_1}. \quad (46)$$

Thus, to *behave like* an irrational ratio over an experiment horizon T_{run} , it suffices that $T_{\text{rec}} \gg T_{\text{run}}$. Integer phase accumulators with $Q = 64$ make q effectively enormous for generic choices of Δ_i .

9 The Stream: environment and evaluation

The intended setting is an open-ended sensory stream: the system does not know group labels, tasks, or episode boundaries. It observes a sequence of transient signals with finite duration; signals ring down rather than disappearing abruptly, and the topology evolves continuously.

Reproducible artifact generation. All paper artifacts referenced in this section are generated by a single command:

```
python3 tmp/rez/paper_artifacts.py.
```

It writes figures/tables into `tmp/rez/artifacts/`, which this L^AT_EX file includes automatically via `\IfFileExists`.

9.1 Core stream metrics

Carrier coherence. Track $\text{coh}(k)$ in (14) and the normalized score D_k .

Synchronization order parameter. A global phase order parameter detects collapse into full synchrony:

$$R(t) = \left| \frac{1}{N} \sum_{i=1}^N e^{i\phi_i(t)} \right| \in [0, 1]. \quad (47)$$

Large R indicates global alignment (often undesirable if it destroys representational diversity).

Sparsity and description length proxies. Track $\text{nnz}(P)$, number of active carriers ($|c_k|$ above threshold), and average row degree d . These are direct proxies for representational complexity.

Compression metric (operationalized). To make the thesis “learning as compression” empirically testable, we operationalize compression using a simple description-length proxy:

$$L_{\text{comp}} = \text{nnz}(P) + M_{\text{active}}, \quad (48)$$

where $\text{nnz}(P)$ is the number of nonzero presences and M_{active} is the number of active carriers. This is intentionally crude but *computable* and captures the two dominant storage/complexity terms in this framework.

Attractor stability. In the stream setting, stability is measured by churn: carrier birth (mitosis) rate, dissolution rate, and the distribution of carrier lifetimes.

9.2 Stream report (auto-generated)

Stream run. We simulate an open-ended sensory stream for 60 s (seed=0, $\Delta t=0.005$ s). Mean $N=4.7$, mean $M=0.9$, mean $\text{nnz}(P)=6.9$, mean $L_{\text{comp}}=7.8$. Mitoses=1, dissolutions=22.

Table 1: Stream summary metrics (time-averaged).

\bar{N}	\bar{M}	$\overline{\text{nnz}(P)}$	$\overline{L_{\text{comp}}}$	Mitoses	Dissolutions
4.7	0.9	6.9	7.8	1	22

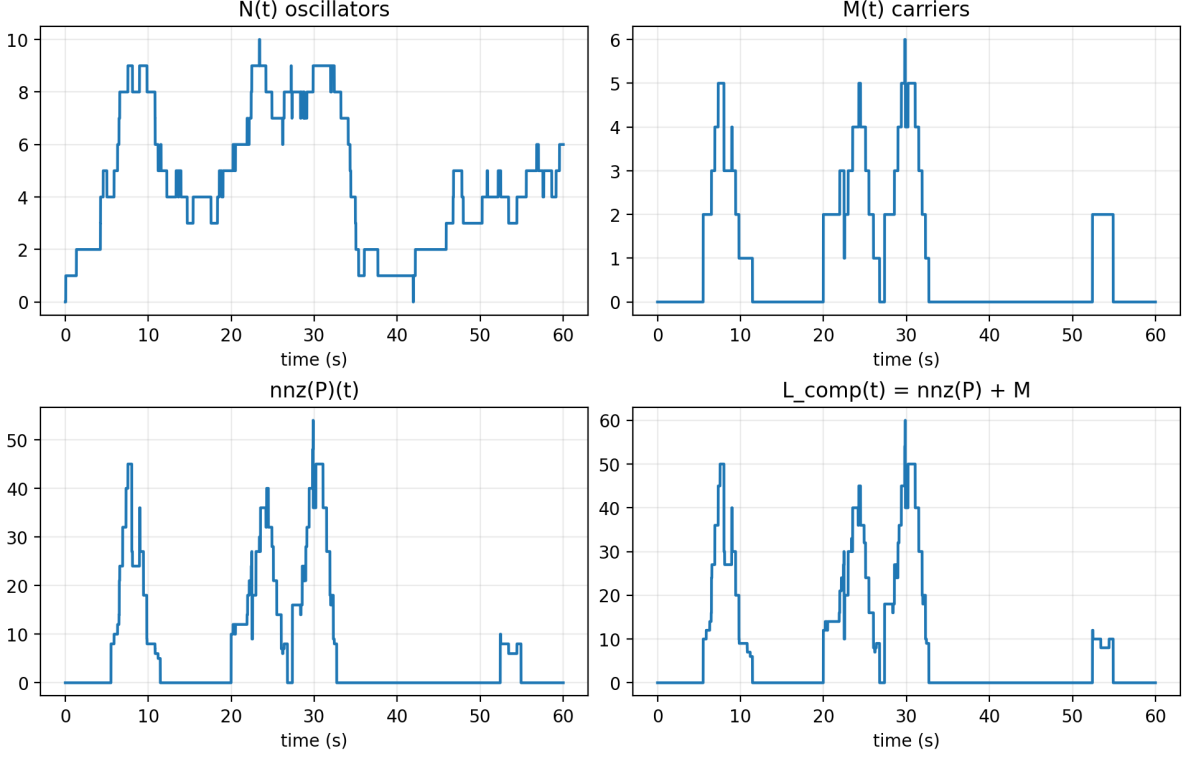


Figure 1: Stream metrics over time: $N(t)$, $M(t)$, $\text{nnz}(P)$, and $L_{\text{textcomp}}(t)$.

9.3 Known failure modes

The following failure modes are expected and have been observed in prototype simulations:

- **Global synchrony collapse:** if coupling is too strong relative to detuning, $R(t) \rightarrow 1$ and representational diversity is lost. Operational fix: reduce κ , increase detuning/noise, or constrain presences.
- **Over-mitosis / carrier proliferation:** aggressive instability thresholds or short persistence windows can create a cascade of mitoses. Operational fix: larger T_{div} and/or a refractory timescale after mitosis.
- **Under-mitosis / failed separation:** if interference does not persist (D_k does not drop below baseline), the system may keep a single mixed carrier. Operational fix: increase detuning between groups or adjust θ_{div} and T_{div} .

9.4 Limitations and scope

The following directions are intentionally out of scope for this paper:

- reinforcement learning and full agentic credit assignment,
- large-scale supervised benchmarks and competitive accuracy claims,
- biological modeling or claims of biological realism,
- hardware-specific implementations beyond the compute scaling discussion.

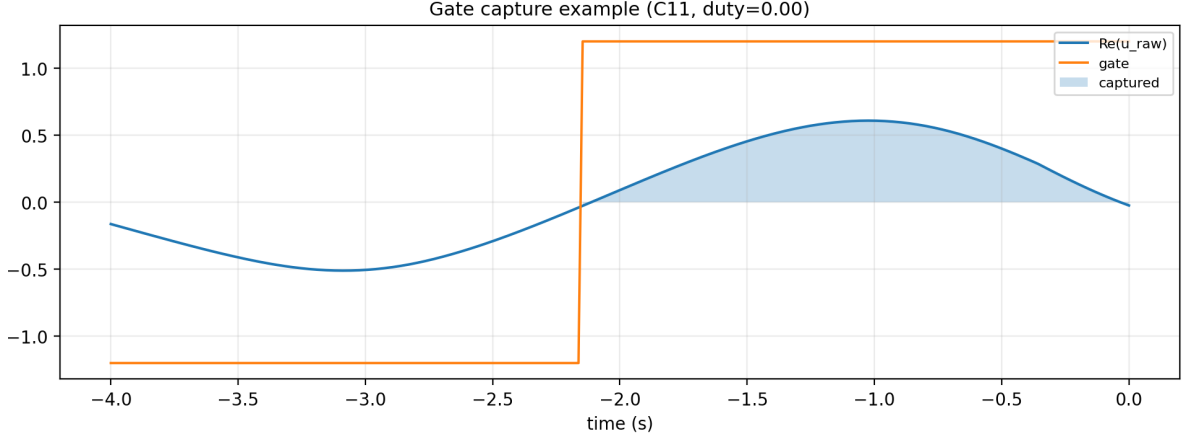


Figure 2: Gate capture visualization: carrier gate (square wave) and the portion of the composite drive captured while the gate is open.

10 Summary

This revised draft resolves several critical missing components for a paper-ready and experiment-ready formulation:

- **Compute requirements:** explicit $O(\text{nnz}(P))$ per-step scaling, sparse simulation loop, GPU-friendly operations, and reduction strategies (rotating frame, multi-rate, quasi-steady carriers).
- **Readout:** concrete output layer designs (output carriers/oscillators), discrete and probabilistic decoding, and training via clamping and three-factor updates.
- **ϵ in finite precision:** reinterpretation as detuning floor, plus implementable strategies: deterministic detuning bounds, integer phase accumulators, and controlled noise/dither.
- **Additional math:** complex carrier formulation, weighted coherence statistics, random-phase baseline constant, and reduction to an effective low-rank Kuramoto network.

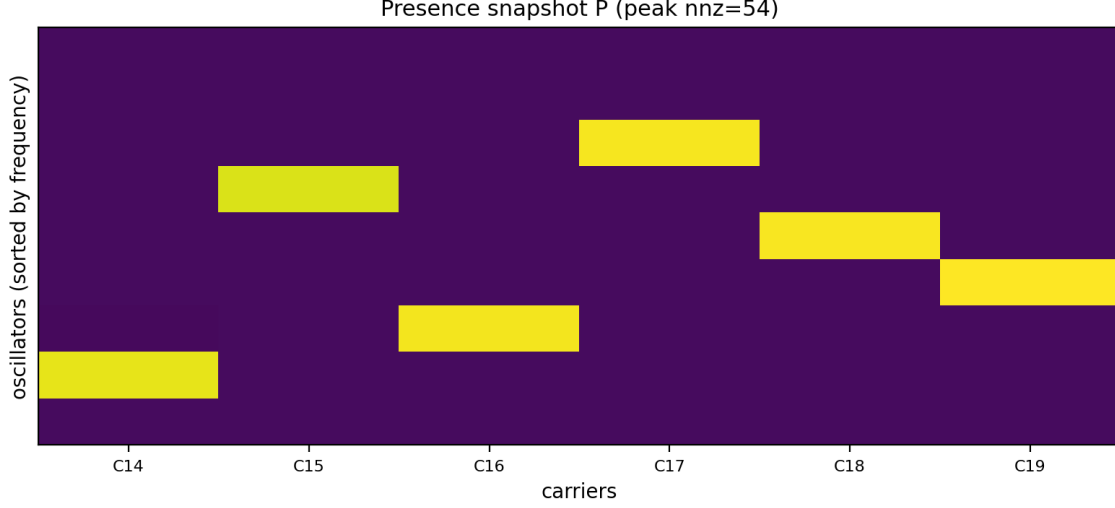


Figure 3: Presence matrix snapshot P (heatmap), showing emergent sparsification by bond snapping.

A Notation quick reference

Symbol	Meaning
N	number of oscillators
M	number of carriers
M_{active}	number of active carriers (above a threshold)
ϕ_i	phase of oscillator i
ω_i	natural frequency of oscillator i
A_i	amplitude of oscillator i
$z_i = A_i e^{i\phi_i}$	complex phasor of oscillator i
c_k	complex amplitude of carrier k
P_{ik}	presence of oscillator i on carrier k
$\text{nnz}(P)$	number of nonzeros in the presence matrix
$u_k = \sum_i P_{ik} z_i$	carrier drive
$g_i = \sum_k P_{ik} c_k$	back-influence on oscillator i
γ_k, β_k	carrier damping and saturation
κ	coupling strength
$\text{coh}(k)$	carrier coherence score (Eq. (14))
D_k	normalized coherence score vs. random-phase baseline
$R(t)$	global phase order parameter
L_{comp}	description-length proxy (Eq. (48))
\hat{f}_i, \hat{f}_k	observable frequency estimates from phase increments (Hz)

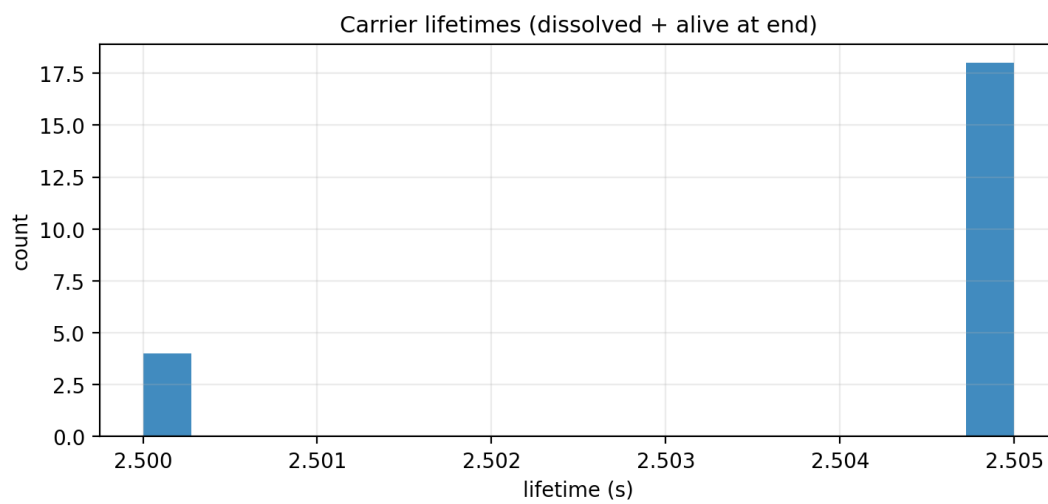


Figure 4: Carrier lifetime distribution under mitosis and metabolic dissolution.