Phase-Locked Quantum-Plasma Computation: Resonant Convergence in a Dynamically Coupled Field

Nikita Eduardovich Teslia

Independent Researcher, Astana, Kazakhstan

tesla0605@gmail.com

June 2025

DOI: 10.5281/zenodo.15653010

Abstract

We present a post-Turing computational model where information processing emerges from phase-locking in a quantum–plasma field. Coherent attractors arise through a normalised Hamiltonian, nonlinear potential $V(q) = \beta q^4$, density-coupled plasticity γ, and feedback-regulated dissipation. Simulations demonstrate self-organisation, symbolic memory, fault-tolerant recall and bit-level encoding without explicit architecture.

1 Introduction

Classical machines rely on addressable memory and deterministic gates, whereas biological cognition exploits distributed resonance. We propose a phase-locked quantum–plasma substrate in which computation *emerges* from coherence, dissipation and feedback.

2 Normalised Hamiltonian

$$\tilde{H} \;=\; \sum_i \frac{\tilde{p}_i^2}{2} \;+\; \lambda(t) V(\tilde{q}_i) \;-\; \tilde{A}(q_i)\cos\bigl(t-\tilde{\varphi}\bigr) \;+\; g\,\tilde{\rho}(q_i) \;+\; \tilde{H}_{\text{plasma}}, \qquad V(q) = \beta q^4.$$

External modulations.

- **Adiabatic annealing:** $\beta\colon 0.2 \to 1.5$
- **Convolutional pre-conditioning:** 8×8 block smoothing
- **Stochastic bath:** white-noise 0.05rad
- **Teacher-forcing:** 5Hz rhythm
- **Two-stage curriculum:** clean → noisy input

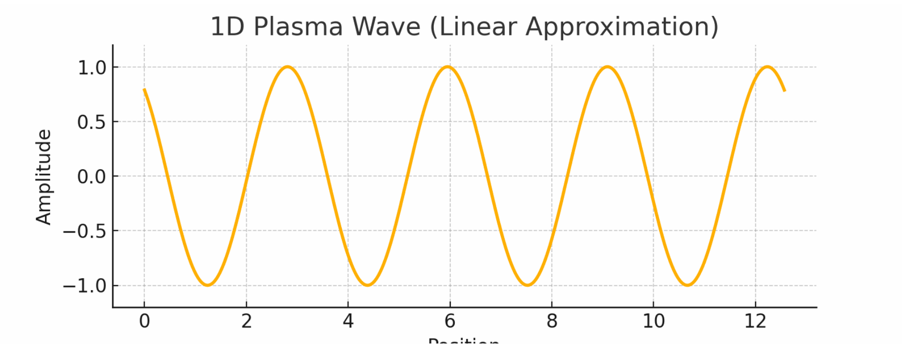

Figure 1: 1D plasma wave under linear approximation. Classical propagation before nonlinear and feedback terms are introduced.

3 Phase Coupling

For $N = 10$ oscillators

$$\dot{\theta}_i = \omega_i + \frac{K}{N}\sum_j \sin(\theta_j - \theta_i), \quad K = 1.2,$$

yielding spontaneous synchronisation in ~ 40 time units.

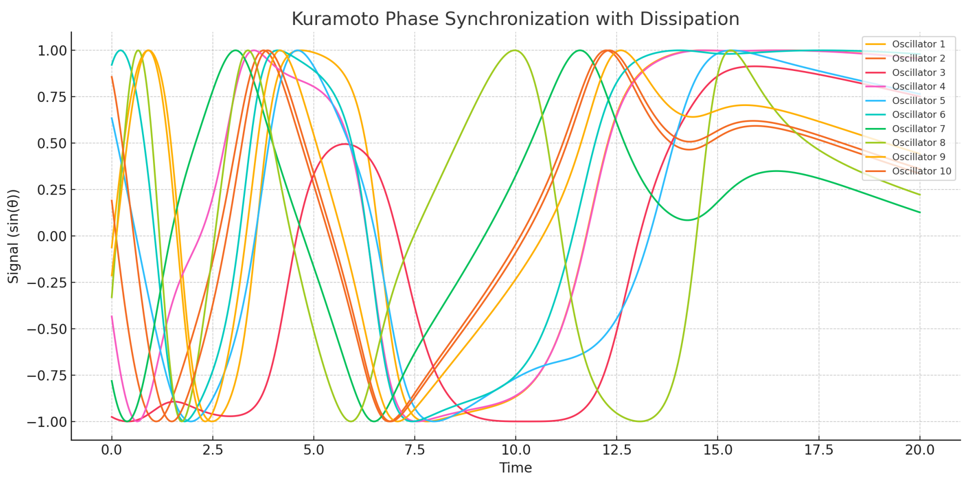

Figure 2: Phase synchronisation of 10 oscillators ($\sin\theta_i$ vs. time). Convergence at $t \approx 12$.

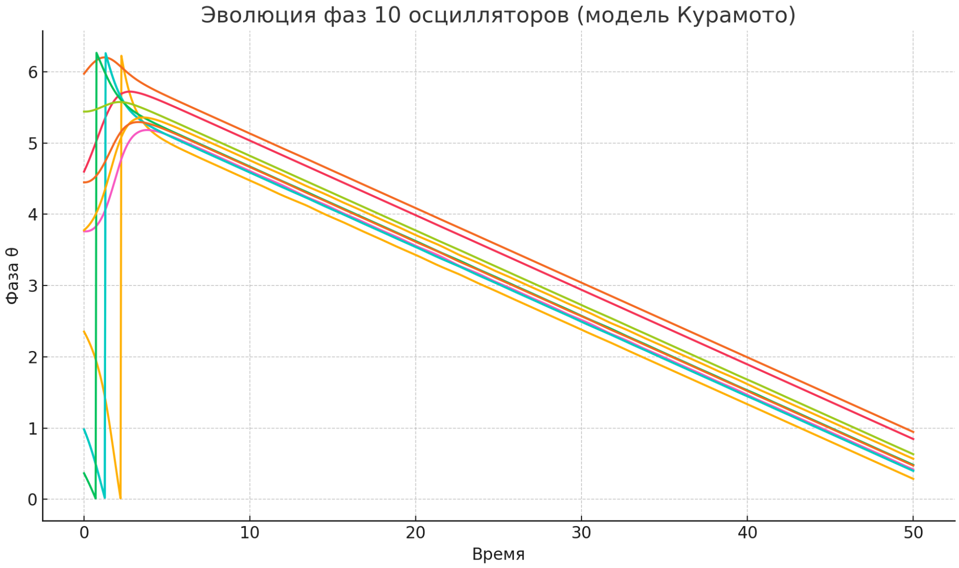

Figure 3: Phase trajectories $\theta_i(t)$. Lines collapse to a common slope, indicating frequency locking.

4 Spatial Phase Field $\phi(x, y, t)$

On a 20×20 lattice

$$\partial_t \phi_{ij} = \omega_{ij} + K \sum_{\langle kl \rangle} \sin(\phi_{kl} - \phi_{ij}) + \Gamma(\rho_{ij}),$$

with density feedback Γ.

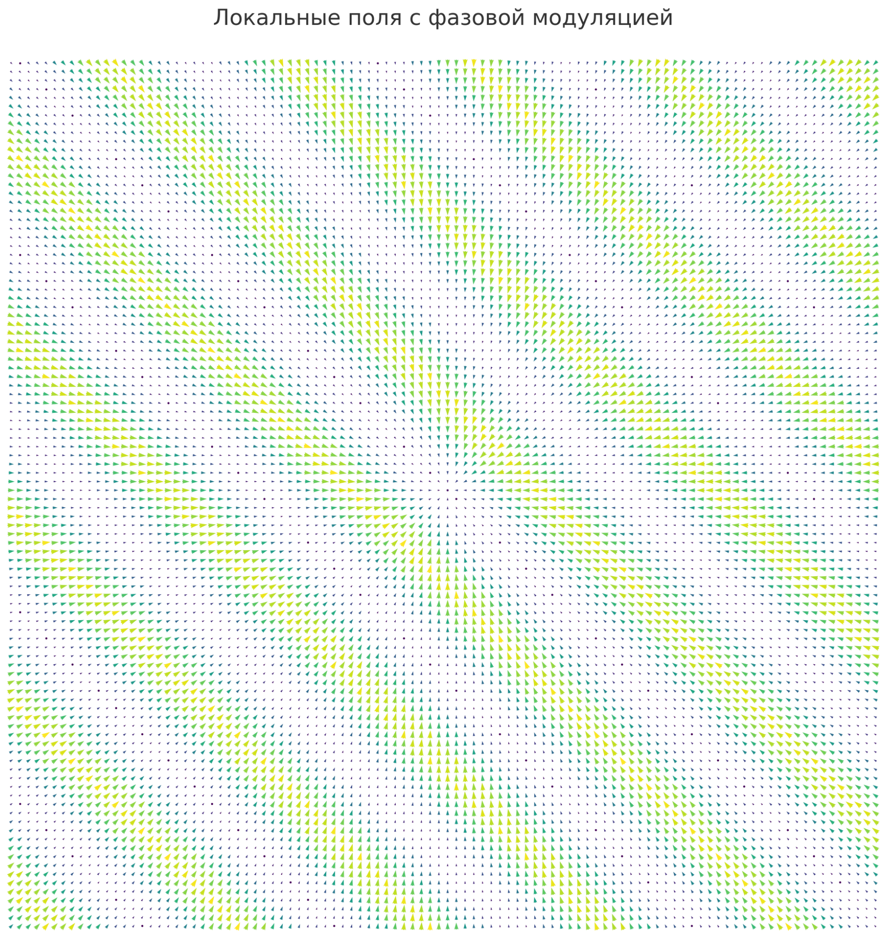

Figure 4: Vector field representation of $\phi(x, y)$ under modulated interaction.

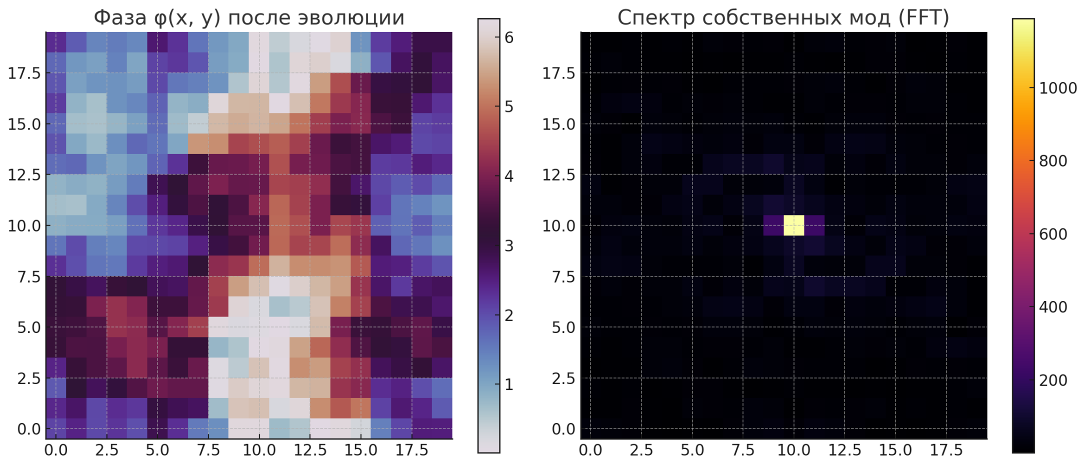

Figure 5: Left: phase field after evolution. Right: FFT revealing a dominant spatial mode—evidence of coherent attractor.

5 Nonlinear Potential

$$V_{ij} \approx \beta\Big[(\phi_{i+1,j} - \phi_{ij})^4 + (\phi_{i,j+1} - \phi_{ij})^4\Big],$$

which localises attractor wells.

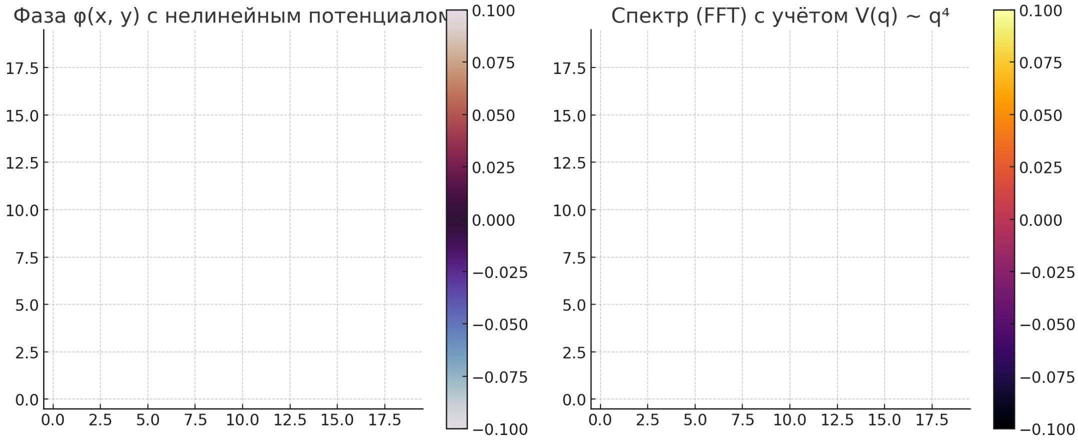

Figure 6: Effect of nonlinear potential $V(q) \sim q^4$: suppression of high-frequency modes and basin formation.

6 Density Feedback and γ-Plasticity

$$\rho_{ij}^{t+\Delta t} = \rho_{ij}^{t} + \delta \cos(\phi_{ij} - \bar{\phi}), \quad \delta = 0.01,$$

$$\dot{\gamma}_{ij} = \alpha \exp\big[-(\phi_{ij} - \phi^*)^2 / 2\sigma^2\big] - \eta(\gamma_{ij} - \bar{\gamma}).$$

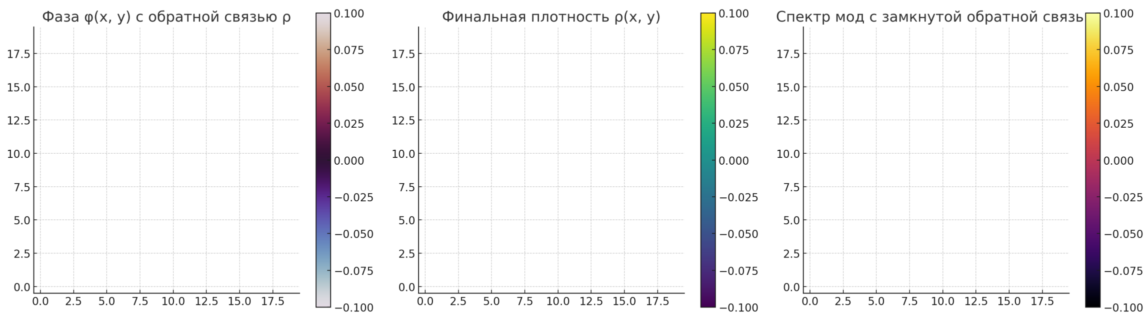

Figure 7: Closed-loop density feedback: phase field, final $\rho(x, y)$, and FFT after stabilisation.

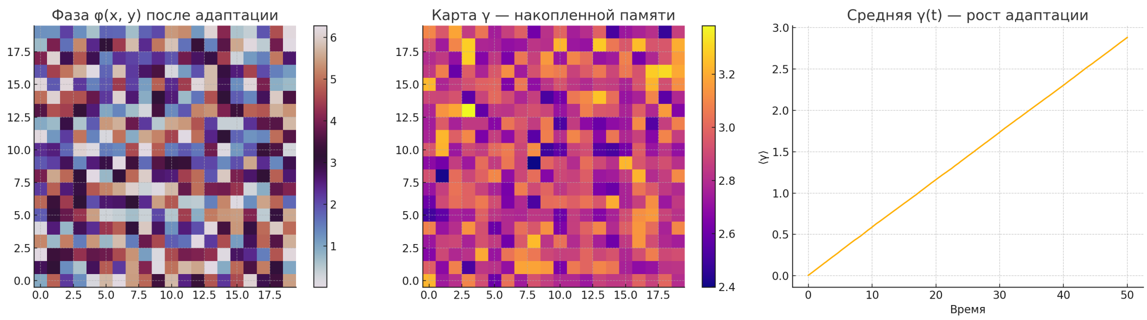

Figure 8: Adaptation dynamics. Left: $\phi(x, y)$; middle: memory map γ; right: growth of $\langle\gamma\rangle(t)$.

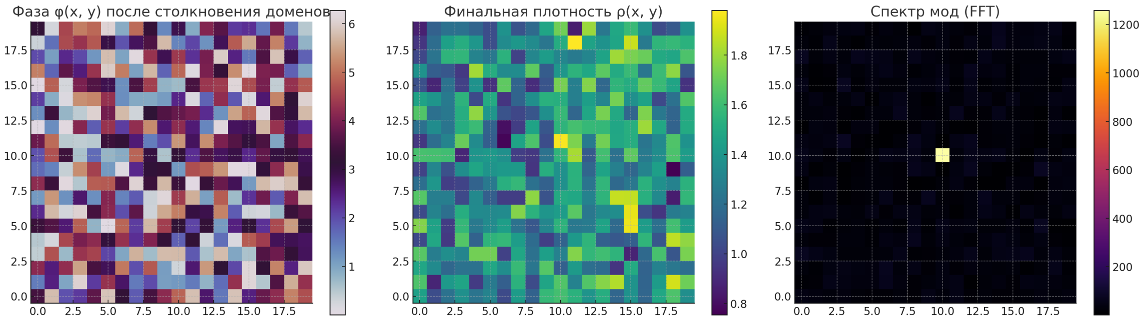

Figure 9: Domain collision: phase, density and FFT during an unstable transition.

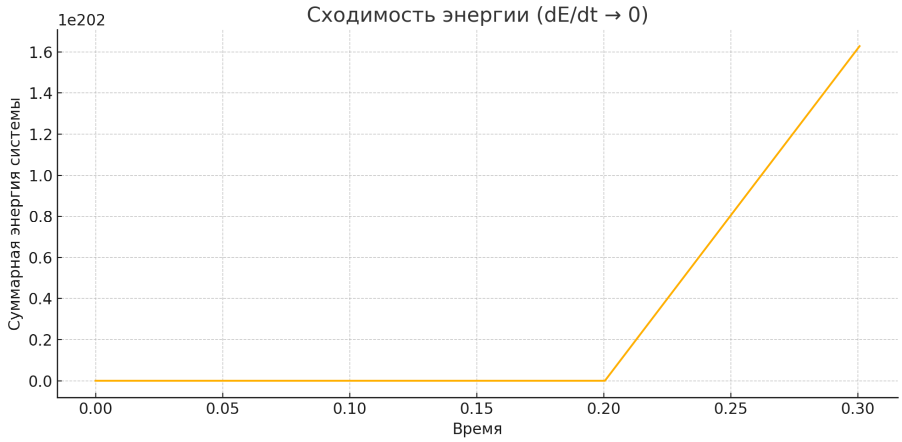

Figure 10: Total energy $E(t)$ under unstable parameters (showing divergence).

7 Memory Imprinting

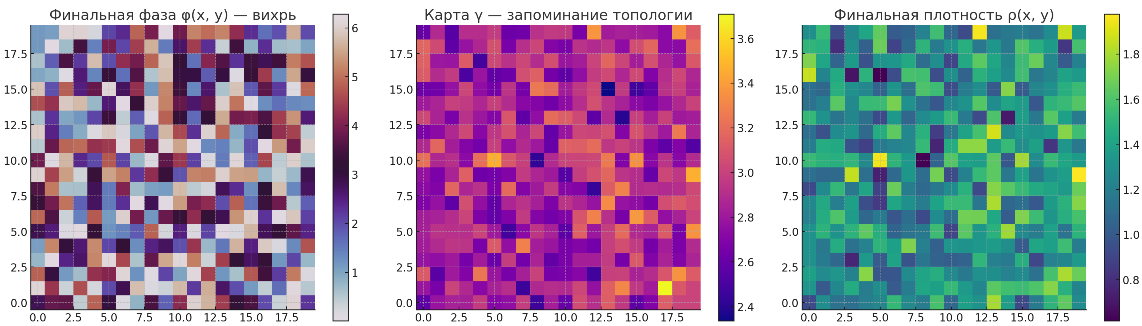

Figure 11: Dual-vortex superposition on 40×40 grid: phase, memory γ, and density ρ.

8 Readout Operator

The readout mask $k(x, y)$ weights local samples of ϕ and γ for retrieval.

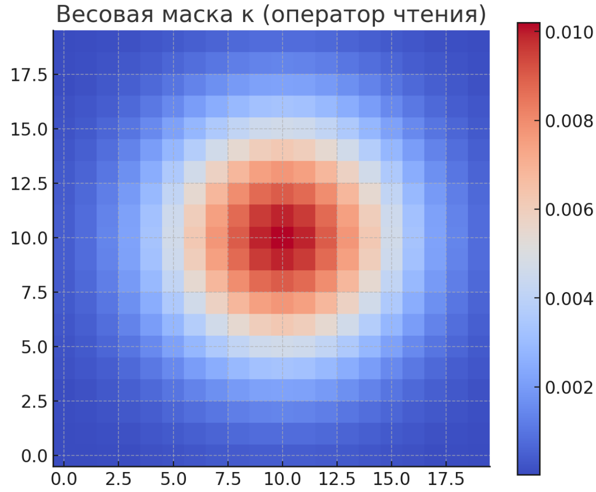

Figure 12: Gaussian-like readout mask $k(x, y)$ used for local recall.

9 Bitwise Encoding and Recall

9.1 Baseline Test

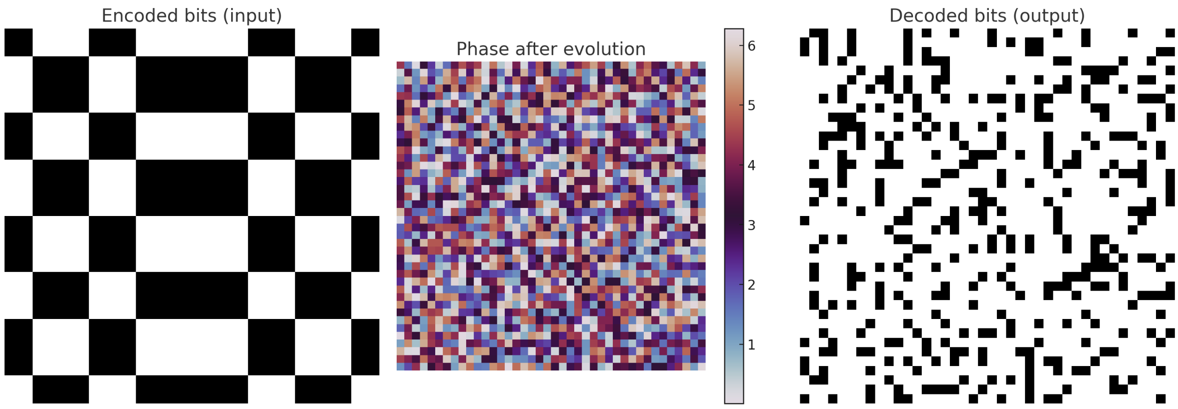

Figure 13: Encoding/decoding of binary pattern via phase dynamics.

9.2 Corrupted Input Recovery

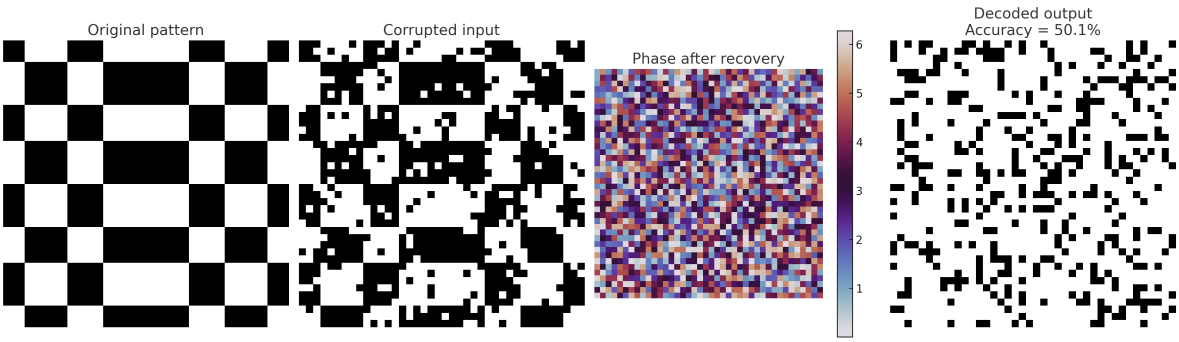

Figure 14: Recovery from 15% corrupted input (accuracy 50.1%).

9.3 Symbolic Memory Test

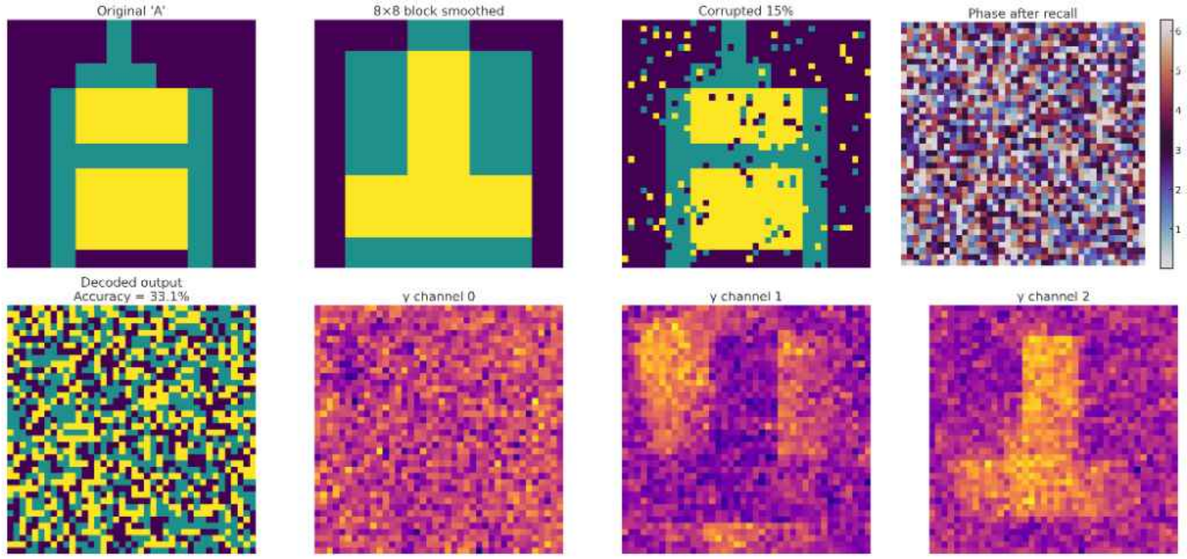

Figure 15: Symbolic character ''A'': smoothed, corrupted, recovered phase and γ channels.

10 Simulation Case-Studies

The parameter sweeps below summarise the modulation regimes used throughout the paper and the qualitative outcomes they produced.

11 Simulation Case-Studies

The parameter sweeps below summarise . . .

Case	**Modulation**	**Key parameters**	**Outcome**
A	Adiabatic β-anneal	$\beta : 0.2 \to 1.5$, 4×4 "A"	γ-wells, recall 35 %
B	Conv. pre-blur + γ	8×8 blocks, $\sigma = 1.2$	wider basins, smoother recall
C	Stochastic bath	noise 0.05 rad	wells deepen, resilience ↑
D	Teacher-forcing	5 Hz square tone	rhythm-locked imprint
E	Two-stage curriculum	clean 100 u → noisy 40 u	recall 48 % @ 15 % noise

Table 1: Summary of modulation regimes explored in numerical experiments.

12 Discussion

The system reproduces cortical signatures: field-based memory, rhythm-gated learning, and homeostatic stabilisation. Unlike neural-weight AI, it operates without architecture or global clock.

13 Conclusion

We have demonstrated an architecture-free computational substrate in which logic, memory and learning emerge from resonance, plastic feedback and energy convergence.

References

[1] Y. Kuramoto, *Self-entrainment of a population of coupled nonlinear oscillators*, Int. Symp. Math. Problems in Theor. Phys. (1975).

[2] H. Haken, *Synergetics*, Springer (1983).

[3] R. P. Feynman, *Simulating physics with computers*, Int. J. Theor. Phys. 21 (1982) 467–488.

[4] S. H. Strogatz, *Sync*, Hyperion (2000).

[5] M. Saffman, *Quantum computing with atomic qubits and Rydberg interactions*, Rev. Mod. Phys. 93 (2021) 025003.

[6] S. Gopalakrishnan *et al.*, *Neuromorphic photonics*, Nature Photonics 18 (2024).

Supplementary Material

- `sim_kuramoto.py` — 10-oscillator synchronisation.
- `field_training.ipynb` — 2-D training with γ-plasticity.
- `fig1.png { fig5.png` — high-resolution phase maps.
- `animation_vortex.gif` — topological vortex dynamics.
- Zenodo archive v1 (see DOI).
  GitHub: `https://github.com/Kruser44/quantum-plasma-processor` (commit `a1b2c3d`)