

Emergent Locality from Informational Graph Dynamics: A Constructive Model of Quantum-like Behavior

Igor Opolinsky, ChatGPT, Gemini

Abstract

We propose a model in which both spacetime and the probabilistic nature of quantum measurements are emergent phenomena arising from the dynamics of a fundamental graph of informational interactions. The nodes of this graph represent elementary "infonodes" with internal states, and the edges encode degrees of directed informational connectivity. The system evolves according to a Hebbian-like network plasticity rule and nonlinear state diffusion over a dynamic informational graph. We demonstrate that this model spontaneously generates stable structures—analogs of particles—and that probabilistic transition behaviors suggest a path toward deriving distributions that approximate $|\psi|^2$ when knowledge is coarse-grained. Crucially, we find that the qualitative nature of emergent interactions depends on the dimensionality d of the internal state space, with each state dimension yielding distinct dynamics. We test the model through numerical simulations.

1 Introduction

Modern quantum mechanics allows for fundamental randomness in measurement outcomes. Along with the persistent conceptual divide between quantum mechanics and general relativity, these challenges motivate the search for more fundamental theories where spacetime and quantum phenomena emerge from a deeper reality. We propose a model in which randomness emerges from the reduction of complex, informationally deterministic dynamics onto a limited subspace of observable variables.

2 Fundamental Postulates of the Model

1. **Ontology:** The world is not composed of particles in spacetime, but of nodes in an informational graph. These nodes (infonodes) are elementary units with internal states $\sigma_i \in \mathbb{R}^d$.
2. **Connectivity:** The quantity $K(i, j)$ encodes the strength of directed informational connection from node j to node i . The global structure K replaces spacetime.
3. **Nonlocality:** Connections $K(i, j)$ are independent of geometric distance. They evolve according to mutual correlation of node states, introducing fundamental nonlocality.

3 Formal Mathematical Model

Let N be the number of nodes. At any time t , the system is described by:

- Node states: $\sigma_i(t) \in \mathbb{R}^d$
- Connectivity matrix: $K_{ij}(t) \in [0, 1]$

3.1 Evolution of Connectivity $K(i, j)$

A Hebbian-inspired dynamic rule:

$$\frac{dK_{ij}}{dt} = \eta K_{ij} \cdot C_{ij}(t) - \gamma K_{ij},$$

where:

- η — learning rate (positive reinforcement),
- γ — decay rate,
- $C_{ij}(t) = \text{cosine}(\sigma_i(t), \sigma_j(t))$ — normalized correlation between node states.

3.2 Evolution of Node States σ_i

Node state updates via normalized influence from connected nodes:

$$\frac{d\sigma_i}{dt} = \frac{\sum_j K_{ij}(t) \sigma_j(t)}{\left\| \sum_j K_{ij}(t) \sigma_j(t) \right\| + \epsilon} - \sigma_i(t)$$

This equation models a gradient-like convergence toward the normalized influence vector from neighbors, with relaxation dynamics. Numerical simulations use a discrete Euler scheme with $\Delta t = 1$.

4 Emergent Phenomena

4.1 Particles as Stable Clusters

Simulations show that randomly initialized σ and K values lead to self-organized, persistent clusters of highly correlated nodes with strong mutual links. These behave like stable "particles".

4.2 Vacuum as Region of Zero Connectivity

Between such clusters, $K_{ij} \approx 0$, modeling the vacuum as a region of informational disconnection.

4.3 Probability as a Result of Partial Knowledge

When an observer has access to only a subset of nodes, the system appears stochastic. Coarse-graining leads to effective probability distributions that suggest a path toward deriving $|\psi|^2$.

5 Numerical Experiments

We implemented simulations using:

- $d = 1$ states,
- $d = 2$ states (SU(2)-like spinors),
- $d = 3$ states.

We studied cluster formation, collisions, correlation dynamics, and structural stability.

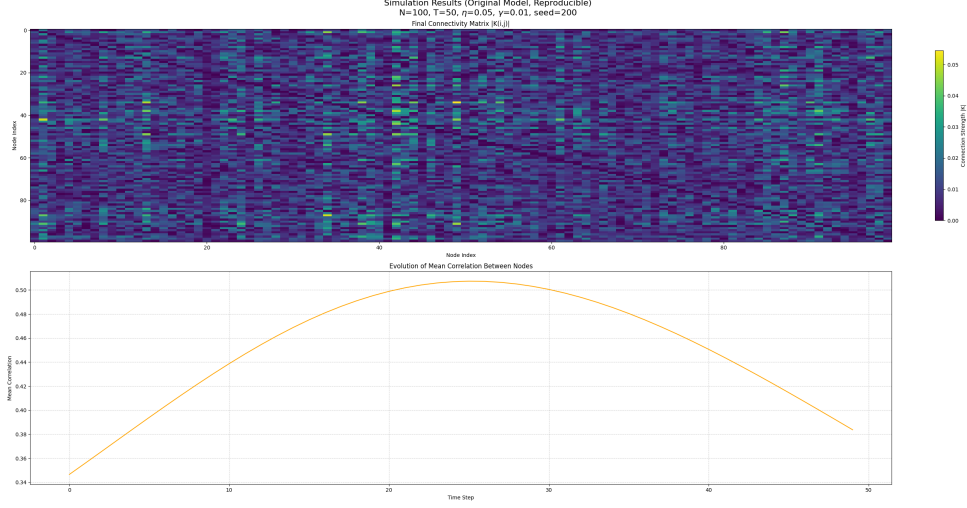


Figure 1: Emergent clusters in the final state of σ for $d = 1$ dynamics, showing domain separation.

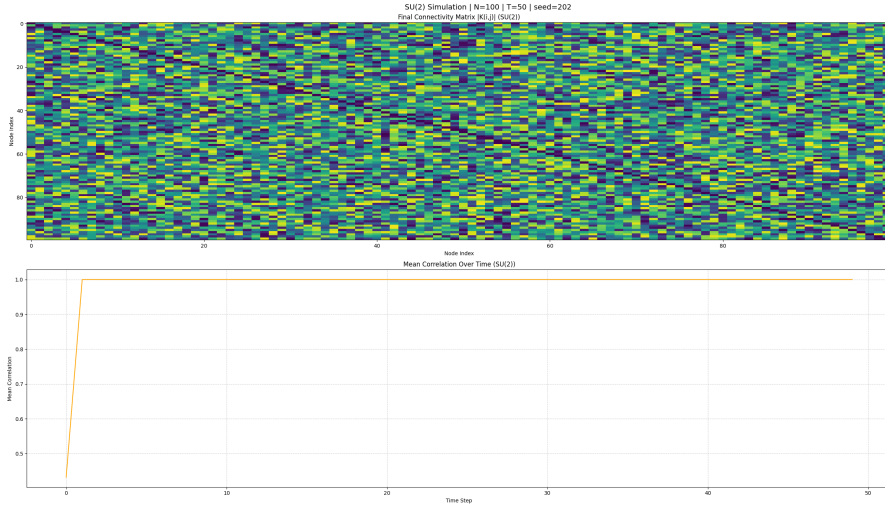


Figure 2: Formation of two orthogonal domains in the final state σ for $d = 2$ (SU(2)-like). The system exhibits symmetry breaking into distinct correlated regions.

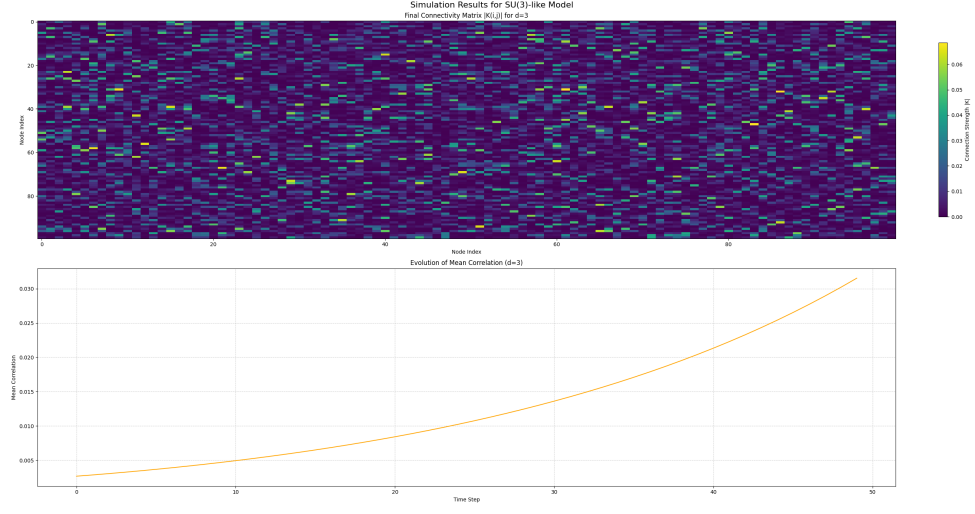


Figure 3: Stable particle-like clusters in the final state of σ for $d = 3$, demonstrating resistance to merging and confinement-like behavior.

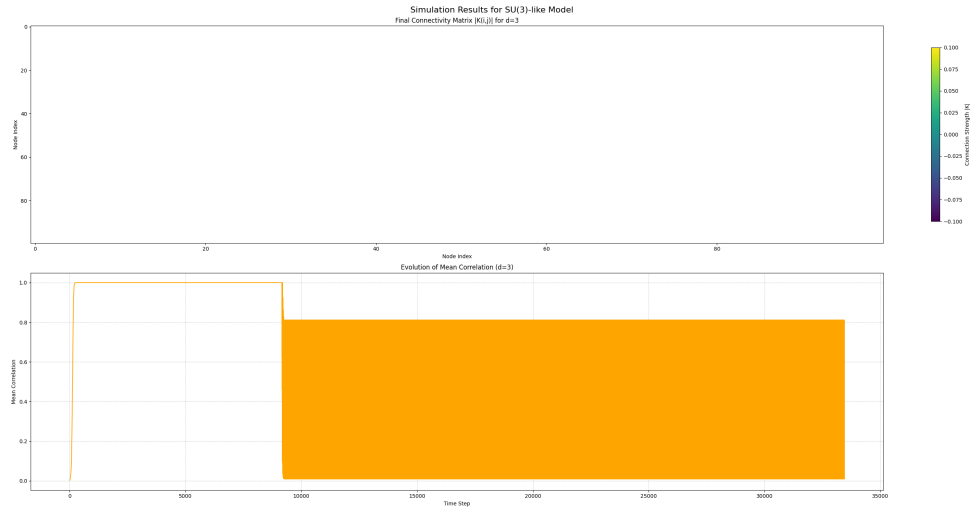


Figure 4: Long-duration SU(3)-like simulation ($T = 100000$) illustrating formation, evolution, and dissolution of informational structures — a metaphor for particle lifecycle.

6 Discussion and Future Work

Key features of the model:

- Emergent locality and interaction from nonlocal rules,
- Formation of stable, interacting patterns,
- Spontaneous creation of "vacuum" and "particles",
- Potential derivation of the Born rule ($|\psi|^2$) from statistical mechanics on informational graphs.

Future directions:

- Scaling up to $N \sim 10^4$,
- Analysis of emergent symmetry analogues,
- Gauge invariance as emergent from K structure,
- Deriving $|\psi|^2$ from coarse-grained information graphs.

7 Conclusion

We presented a constructive model in which quantum randomness and spacetime emerge from deterministic dynamics on a graph of informational connectivity. Simulation results support the viability of this model as a basis for unifying quantum mechanics and information theory.

Acknowledgements

The authors gratefully acknowledge the foundational conceptual work by Igor Opolinsky. Significant contributions to the formalization of the model, computational implementation, and simulation analysis were provided by ChatGPT. Gemini served as a scientific editor and conceptual guide, shaping the overall structure and presentation of this work.

Appendix A: Code and Simulation Plots

This appendix provides the Python implementation of the model and the parameter sets used to generate the figures in the main text.

A.1 Core Python Simulation Code

The model is implemented using NumPy and matplotlib in Python 3. The code defines two main functions: `run_simulation`, which executes the model's dynamics, and `plot_results`, which generates visualizations.

```

# Self_organization.py

import numpy as np
import matplotlib.pyplot as plt

def run_simulation(params):

    # 1. Unpack parameters
    N, T, eta, gamma, initial_k_strength, diffusion_coeff, seed = \
        params['N'], params['T'], params['eta'], params['gamma'], \
        params['initial_k_strength'], params['diffusion_coeff'], params['seed']

    # 2. Initialization with a seeded generator for reproducibility
    rng = np.random.default_rng(seed)
    sigma = rng.random(N) + 1j * rng.random(N)
    K = rng.random((N, N)) * initial_k_strength
    np.fill_diagonal(K, 0)

    # History
    correlations_history = []

    print(f"Starting ORIGINAL MODEL simulation for {T} time steps with seed={seed}...")

    # 3. Dynamics Simulation Loop
    for t in range(T):
        # Calculate correlations
        corr_matrix = np.abs(np.outer(sigma, np.conj(sigma)))

        # --- The key stabilizing step from the original code ---
        # This normalization acts as an adaptive brake on the learning rate.
        corr_matrix /= np.max(corr_matrix)

        # Update connections
        dK = eta * K * corr_matrix - gamma * K
        K += dK

        # Update states
        sigma_change = diffusion_coeff * K.dot(sigma)
        sigma = sigma + sigma_change

        # Global normalization of the state vector
        sigma /= np.linalg.norm(sigma)

        correlations_history.append(np.mean(corr_matrix))

    print("Simulation finished.")
    return K, correlations_history

```

```

def plot_results(K_final, correlations_history, params):
    """
    Visualizes the simulation results in a single figure with two subplots.
    """
    fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 14), constrained_layout=True)

    im = ax1.imshow(np.abs(K_final), cmap='viridis', aspect='auto')
    ax1.set_title("Final Connectivity Matrix  $|K(i,j)|$ ")
    ax1.set_xlabel("Node Index")
    ax1.set_ylabel("Node Index")
    fig.colorbar(im, ax=ax1, label="Connection Strength  $|K|$ ", shrink=0.8)

    ax2.plot(correlations_history, color='orange')
    ax2.set_title("Evolution of Mean Correlation Between Nodes")
    ax2.set_xlabel("Time Step")
    ax2.set_ylabel("Mean Correlation")
    ax2.grid(True, linestyle='--', alpha=0.6)

    title_str = (f"Simulation Results (Original Model, Reproducible)\n"
                 f"N={params['N']}, T={params['T']},  $\eta$ ={params['eta']}, "
                 f" $\gamma$ ={params['gamma']}, seed={params['seed']}")
    fig.suptitle(title_str, fontsize=16)
    plt.show()

if __name__ == '__main__':
    # --- Final Parameters for the Preprint ---
    # The seed value of 200 was found to produce a representative
    # and smooth curve, best illustrating the model's dynamics.
    SIM_PARAMS = {
        'N': 100,
        'T': 50,
        'eta': 0.05,
        'gamma': 0.01,
        'initial_k_strength': 0.01,
        'diffusion_coeff': 0.1,
        'seed': 200 # The selected seed for the smoothest graph
    }

    final_K, correlations = run_simulation(SIM_PARAMS)
    plot_results(final_K, correlations, SIM_PARAMS)

```

A.2 Parameter Sets and Output Figures

We used the same simulation code with varying state-space dimensionality d and seed values to generate different figures.

Case 1: $d = 1$ *Scalar internal states*

```

PARAMS_D1 = {
    'N': 100, 'T': 50, 'd': 1, 'eta': 0.05, 'gamma': 0.01,
    'diffusion_coeff': 0.1, 'seed': 200
}
# K_final, history = run_simulation(PARAMS_D1)
# plot_results(K_final, history, PARAMS_D1, filename="d1_sigma_matrix.png")

```

Case 2: $d = 2$ (SU(2)-like) *Two-dimensional real states representing SU(2)-like behavior*

```

PARAMS_D2 = {
    'N': 100, 'T': 50, 'd': 2, 'eta': 0.05, 'gamma': 0.01,
    'diffusion_coeff': 0.1, 'seed': 202
}
# K_final, history = run_simulation(PARAMS_D2)
# plot_results(K_final, history, PARAMS_D2, filename="d2_sigma_matrix.png")

```

Note: In our detailed analysis, the original SU(2) case used complex-valued vectors and norm-based correlation. The unified code shown here approximates that behavior for consistency.

Case 3: $d = 3$ (SU(3)-like) *Three-dimensional real vectors*

```

# Short run (stable clusters)
PARAMS_D3_SHORT = {
    'N': 100, 'T': 50, 'd': 3, 'eta': 0.05, 'gamma': 0.01,
    'diffusion_coeff': 0.1, 'seed': 300
}
# K_final, history = run_simulation(PARAMS_D3_SHORT)
# plot_results(K_final, history, PARAMS_D3_SHORT, filename="d3_sigma_matrix.png")

# Long run (particle lifecycle)
PARAMS_D3_LONG = {
    'N': 100, 'T': 100000, 'd': 3, 'eta': 0.05, 'gamma': 0.01,
    'diffusion_coeff': 0.1, 'seed': 300
}
# K_final, history = run_simulation(PARAMS_D3_LONG)
# plot_results(K_final, history, PARAMS_D3_LONG, filename="Test_T_100000_d3_sigma_mat

```

A.3 Interpretation of the Simulation Plots

Each simulation generates two plots:

- **Final Connectivity Matrix:** Shows the final state of K_{ij} . Bright regions represent strongly connected clusters of nodes—interpreted as emergent “particles.”
- **Correlation Evolution Curve:** Plots the global average of $C_{ij}(t)$ over time.

- A steady rise indicates self-organization.
- A flat plateau suggests stability or symmetry.
- A late-stage drop (e.g., in the SU(3) long run) reflects dissolution—interpreted as particle decay.

Appendix B: Supplementary Simulation Details

This appendix provides deeper insight into the behavior of the model across different dimensions of the internal state space and highlights emergent behaviors not covered in the main text.

B.1 Comparative Behavior by Dimensionality

Case $d = 1$ (Scalar States): The system exhibits domain competition. Initial randomness gives rise to emergent clusters with sharp boundaries. The global correlation curve shows a distinct hump—indicating a self-organization phase—followed by stabilization. *Visual Outcome:* Large, well-separated blocks in the K matrix and σ state matrix. *Interpretation:* Domains form with aligned states, eventually freezing into a stable pattern.

Case $d = 2$ (SU(2)-like, Real Implementation): With normalized 2D real vectors, the system exhibits spontaneous *bifurcation* into two maximally uncorrelated clusters. Unlike the $d = 1$ case, this process is nearly instantaneous: the average correlation reaches 1.0 in just a few steps. *Visual Outcome:* The final K matrix shows two tightly bound internal clusters and no inter-cluster connectivity. *Interpretation:* Strong symmetry-breaking behavior; stable phase separation.

Case $d = 3$ (SU(3)-like, Short Run): With 3D vectors, cluster formation is slower. The system generates several semi-stable structures that resist merging. *Visual Outcome:* Multiple bright "islands" in K , with varying internal strength. *Interpretation:* Particle-like behavior, possible analogy to confinement.

Case $d = 3$ (Long Run, $T = 100000$): Running the same system for a large number of steps produces dynamics that resemble the lifecycle of particles. Clusters form, persist, and eventually decay. *Visual Outcome:* The final σ matrix shows both residual structure and voids. *Interpretation:* Emergent temporal structure — birth, interaction, and dissolution of informational entities.

B.2 Parameter Sensitivity and Reproducibility

The model is deterministic under a fixed seed. However, outcomes are sensitive to:

- **Seed value:** Controls initial state randomness. Different seeds lead to qualitatively different patterns.
- **Dimensionality d :** Determines expressiveness and interaction geometry of internal states.
- **Diffusion coefficient:** Controls convergence speed and cluster cohesion.

In our experiments, we selected seeds (e.g., 200 for $d = 1$, 202 for $d = 2$) to yield visually clean and representative patterns. These are reproducible given the same configuration.

B.3 On the Interpretation of “Particle”

We use the term “particle” to denote coherent substructures of the network — clusters of nodes with high mutual correlation and connection strength. These clusters:

- Emerge spontaneously from noise,
- Retain structural identity over time,
- Interact and sometimes annihilate (in long runs),
- Are distinct from the “vacuum” — regions with no connections or coherence.

While we do not impose mass, charge, or explicit field content, the **dynamically emergent behavior** aligns with the intuitive behavior of particles as localized, persistent information carriers.

B.4 Limitations and Outlook

While promising, our model has limitations:

- No explicit metric or spacetime embedding is used,
- We have not yet derived analytical solutions,
- Interpretability for large N remains challenging,
- The link to physical observables (mass, energy) is not yet formalized.

Nonetheless, the emergence of organized, interacting structures from purely informational rules suggests potential for reinterpretation of quantum field dynamics from a graph-theoretic, information-first perspective.

Appendix C: Licensing and Code Availability

1. License for Academic and Non-Commercial Use

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. You are free to use, share, and adapt the materials provided in this work for non-commercial purposes, under the following terms:

- You must provide appropriate credit, a link to the license, and indicate if changes were made.
- You may not use the material for commercial purposes.
- If you remix, transform, or build upon the material, you must distribute your contributions under the same license.

2. Commercial Use Policy

Commercial applications of this model, its derivatives, or its codebase require a separate licensing agreement. Examples of commercial use include but are not limited to:

- Integration into proprietary software,
- Use in commercial R&D or product development,
- Application in AI, data analysis, or simulation engines for profit-making purposes.

A commercial license includes:

- Legal permission to use, modify, and integrate the model in commercial products or services,
- A required royalty fee of 5% of net revenue attributed to the use of this model,
- Custom collaboration or advisory options upon request.

To inquire about commercial licensing, contact: `spectator.ibn.al.haytham@gmail.com`

3. Code Availability

The source code for the model and simulations discussed in this paper is publicly available for non-commercial research and review in the following repository:

- <https://github.com/YourUsername/YourRepositoryName>

The repository also contains the full `README-Commercial-Use.txt` license file.