String Network Gravity

A Structural Approach to Spacetime Deformation and Black Hole Phase Transitions

Version 4.0

AL_78 | Independent Research Group | June 2026

Abstract

We propose a novel discrete framework for gravitational phenomena based on the concept of *string elements*—edges of a cubic complex whose connectivity and weights constitute the fundamental structure of space. In this model, gravitational effects are modelled without reference to force, curvature, or spacetime manifold—only as a *structural deformation* of the string network induced by the presence of load. We demonstrate that network structures termed *black holes* within this framework correspond to *phase transitions* in the network topology, replacing the classical singularity with a limiting connectivity regime. The *horizon*, as defined within this framework, is a combinatorial boundary where the network ceases to support outward information pathways; this is an autonomous definition within the SNG ontology, not a reinterpretation of the general-relativistic event horizon. We develop the dynamical equations governing network evolution, analyze stationary states and linear stability, and establish formal connections to Regge calculus, graph Ricci flow, spectral graph theory, and percolation theory. The model exhibits area-law scaling of *configurational complexity* S_{conf}—a combinatorial quantity, explicitly distinguished from thermodynamic entropy and referred to as "entropy" below only by loose structural analogy (Section 5.3)—through combinatorial arguments; this is structurally analogous to area-law behaviour in emergent geometric interpretations, but arises from percolation-theoretic properties of the network, not from thermodynamic entropy. **Three-dimensional validation.** GPU-accelerated simulations on a 64^3 cubic lattice with **500 independent realisations per parameter point** yield an area-law coefficient $C = 12.38 \pm 0.06$ ($R^2 = 0.99997$). The proximity to 4π is an empirical constraint on the square-lattice calibration, not a derivation of the Bekenstein–Hawking coefficient. The SNG phase transition is robust across dimensions; the percolation-dominated regime prevails in the strong-field limit in both 2D and 3D. Two-dimensional consistency checks of the horizon scaling law and the area law $N_H \propto r_c^2$ are additionally presented for the screened Poisson limit. **Lattice dependence of the area-law coefficient.** A systematic scan over different 2D lattice types (square, triangular, hexagonal) reveals that the coefficient $C = N_H/r_c^2$ depends on the lattice geometry: $C_{\text{square}} = 12.431 \pm 0.006$, $C_{\text{triangular}} = 18.850 \pm 0.000$, and $C_{\text{hexagonal}} = 18.533 \pm 0.100$. The area law $N_H \propto r_c^2$ holds in all cases, confirming its robustness; the prefactor is a dimension- and geometry-dependent quantity, not a universal constant.

We further demonstrate that the load relaxation process corresponds to a topological phase transition in which the horizon shrinks and eventually vanishes, with the total network state conserved throughout. This constitutes network state conservation in a classical topological system, not a resolution of the quantum black-hole information paradox.

The numerical coefficient $C \approx 4\pi$ is an empirical finding on the square lattice and is not a universal prediction. The robust result is the quadratic scaling $N_H \propto r_c^2$, which holds across all lattice types and dimensions; the prefactor C is geometry-dependent.

Keywords: discrete gravity, string networks, phase transitions, black holes, black hole evaporation, Regge calculus, graph theory, non-singular spacetime, network state conservation

Important Note

This document constitutes a **conceptual and theoretical contribution**. Many of the theoretical assertions herein—particularly those pertaining to continuum limits and the dynamics of phase transitions—remain **working hypotheses rather than established theorems**, notwithstanding the inclusion of mathematical definitions, propositions, and numerical consistency checks. (See the Ontological Contract, Section 0, for the full statement of scope and relation to General Relativity, LQG, CDT, and string theory.)

The authors are independent researchers, operating without institutional affiliation and without recourse to supercomputing facilities or to data from major research consortia (e.g., DESY, Planck, LIGO). All numerical verifications reported herein were conducted on a standard laptop computer. The three-dimensional scan (Section 5.7) was performed on an NVIDIA RTX PRO 6000 Blackwell Server Edition GPU using CuPy, with 500 independent realisations per parameter point. Earlier consistency checks (Appendices A–E) were conducted on a standard multi-core laptop.

All discussion of prospective development—including timelines and computational resources—is **speculative** and does not constitute a commitment. We remain open to collaboration, consultation, and support in any form—research infrastructure, theoretical input, experimental verification, or funding.

We fully acknowledge that our model is Euclidean, lacks Lorentzian signature, and does not include coupling to matter fields. These are features of a conceptual proposal at this stage, not deficiencies of a physical theory.

0. Ontological Contract

How to read this work. This document is not a theory of gravity in the conventional sense. It does not derive, reproduce, amend, or compete with General Relativity. It does not predict gravitational waves, black hole shadows, or perihelion precession. It does not couple to matter fields. It is Euclidean, not Lorentzian. It contains no spacetime manifold at the fundamental level.

What this document is. A self-consistent mathematical framework in which geometric and gravitational analogues emerge from the dynamics of a discrete network. Terms such as "gravity," "horizon," "entropy," "black hole," and "wave" are defined internally within this network ontology. They are structural labels, not claims about physical spacetime.

What this document does. It demonstrates that a minimal combinatorial system—vertices, edges, weights, and a local threshold—can exhibit (i) exponential screening of deformation, (ii) phase transitions with area-law scaling of boundary entropy, and (iii) dynamical restoration of global connectivity. These behaviours are structurally analogous to features of gravitational physics, but they arise from percolation-theoretic and spectral-graph mechanisms, not from curvature or thermodynamic entropy.

What this document does not do. It does not claim that the network is spacetime, that the load parameter is mass, or that the phase transition is gravitational collapse. Such identifications, where offered, are heuristic bridges (Table 9), not foundational equations. The model is advanced as a generative substrate—a conceptual proposal for how geometric structure might emerge from pre-geometric combinatorics—not as a completion of existing physics.

To the reader. If you are looking for a derivation of Einstein's field equations, a quantum theory of matter, or a falsifiable prediction for LIGO or the Event Horizon Telescope, you will not find them here. If you are willing to entertain the possibility that space itself is a dynamical system of relations, and that "gravity" might be a name for structural deformation rather than a fundamental force, the following framework is offered for examination.

Contents

1. Introduction

The present work offers a conceptual model of spacetime structure in which gravity arises from the internal dynamics of the network rather than being postulated as a fundamental interaction. The term 'gravity' in the title denotes a structural property of the network, not a force described by Einstein's field equations. In this work, gravity is treated as an effect of network deformation, not as curvature of spacetime.

The reconciliation of general relativity with quantum mechanics remains one of the central challenges of theoretical physics. While string theory, loop quantum gravity, and other approaches have made significant progress, they often retain the concept of a smooth spacetime manifold at fundamental scales or introduce additional dimensions and symmetries that lack direct observational support.

This work proposes an alternative ontological foundation: **space itself is a network of relational elements** ("strings"), and gravitational phenomena emerge from the dynamical reconfiguration of this network rather than from geometric curvature in a pre-existing manifold. The key conceptual shift is from asking "what moves in spacetime?" to "how does space itself change as a system of relations?"

Our approach is deliberately minimal: we begin with a cubic complex (abstracted from any embedding), assign dynamical weights to its edges, and study how localized "load" parameters induce network deformations. In the strong-field limit, the network undergoes *phase transitions*—topological reconfigurations that replace the classical notion of a singularity with a well-defined structural limit.

A crucial aspect of the present formulation is that the 2D lattice simulations probe one specific regime of the SNG framework: the percolation-dominated regime, where the horizon size is determined primarily by the topological threshold ρ_c rather than by the load amplitude. The percolation-dominated behaviour does not contradict the gravitational interpretation of SNG; rather, it reveals the microscopic mechanism by which gravitational collapse manifests as a structural phase transition. Additional regimes may become accessible in the weak-field limit or with alternative parameter calibrations.

Three-dimensional validation (Section 5.7). To establish dimensional robustness, we performed GPU-accelerated simulations on a 64^3 cubic lattice with **500 independent realisations per parameter point**. The results confirm that the SNG phase transition — including the area law $N_H \propto r_c^2$, the percolation-dominated horizon formation, and the structural interpretation of black holes — is robust in three dimensions. The area law coefficient $C = 12.38 \pm 0.06$ ($R^2 = 0.99997$) is close to 4π; this proximity is an empirical constraint of the square-lattice calibration, not a derivation of the Bekenstein–Hawking coefficient. This is not a preliminary result: it is a statistically robust confirmation on a full 3D lattice with high-performance computing.

The question of whether and how a Lorentzian continuum limit with dynamical matter might emerge from the SNG framework is discussed as an open research programme in Section 8. The present work is deliberately limited to the Euclidean, discrete regime; no claim is made that the current model reproduces General Relativity or any established continuum theory of gravity.

The dynamical relaxation of a localised load in SNG provides a structural analogue of black hole evaporation. In SNG, the black hole is not a geometric object but a topological phase of the network. Its 'evaporation' is therefore a phase transition—a dynamical reconfiguration of edge weights and connectivity flags—rather than a mysterious disappearance of a spacetime singularity. The network state remains conserved because the full configuration $(\mathbf{w}, \boldsymbol{\kappa})$ is retained throughout; only the accessibility of information changes as the network topology evolves. This is network state conservation in a classical topological system. It is structurally analogous to, but does not resolve, the quantum black-hole information paradox, which lies beyond the scope of this classical, Euclidean framework.

2. Fundamental Postulates

Postulate 1 (String Elements). *The fundamental constituents of space are **string elements** $e \in E$, modeled as edges of an abstract cubic complex $\mathcal{C} = (V, E, F, \ldots)$. Each string connects exactly two vertices (0-cells) and carries no intrinsic direction.*

Postulate 2 (Null-Dimensional Vertices). *Vertices $v \in V$ are **purely combinatorial objects** with no internal extension, position, or coordinates. A vertex is defined entirely by the set of incident edges: $v \equiv \{e_1, e_2, \ldots, e_k\}$.*

Postulate 3 (Emergent Metric). *The metric structure of space is secondary, emerging from the weights $\{w_e\}$ assigned to string elements. The distance between vertices is the minimum weighted path length:*

$$d(u, v) = \min_{\gamma : u \to v} \sum_{e \in \gamma} w_e. \tag{1}$$

Because $w_e > 0$ for all $e \in E$, the triangle inequality holds by path concatenation, and $d(u, v)$ is a proper metric on V.

Postulate 4 (Structural Gradient (termed "gravity" within this framework)). *Within the SNG ontology, what is termed **gravity** denotes gradients of network restructuring. This is a primitive relational property, not a force or curvature in any pre-existing geometric manifold. The presence of load at a vertex induces local deformation of incident edge weights, propagating through the network via connectivity constraints.*

Postulate 5 (Phase Transition Limit). *When local network density exceeds a critical threshold ρ_c, the configuration becomes unstable and undergoes a **phase transition** to an alternative organizational regime. This replaces the classical singularity as the limiting behavior of extreme gravitational collapse.*

3. Mathematical Framework

3.1. State Space of the Network

The complete state of the network at time t is given by:

$$\mathcal{C}(t) = \big(V,\, E,\, \mathbf{w}(t),\, \boldsymbol{\kappa}(t),\, \mathbf{m}\big), \tag{2}$$

where:

- V — set of vertices (0-cells);

- E — set of edges/strings (1-cells);
- $\mathbf{w}(t) = \{w_e(t)\}_{e\in E}$, with $w_e > 0$ the effective length of edge e;
- $\boldsymbol{\kappa}(t) = \{\kappa_e(t)\}_{e\in E}$, with $\kappa_e \in \{0, 1\}$ the connectivity flag (1 = active, 0 = severed);
- $\mathbf{m} = \{m_v\}_{v\in V}$, with $m_v \geq 0$ the load parameter at vertex v.

In physical units, the load is dimensionless when expressed as $\mathcal{I} = E/E_P$, where E is the energy of the localized disturbance and E_P is the Planck energy. This is a calibration, not a derivation.

3.2. String Parameters

Each edge e is characterized by a triple $(w_e, \kappa_e, \sigma_e)$:

Definition 1 (Tension Function)**.** *The natural (unstressed) length of an edge is $w_e^{(0)}$. The tension function ϕ measures deviation from this natural state. To avoid a discontinuity, we use a smooth bounded function:*

$$\phi(w_e, w_e^{(0)}) = \tanh\left(\frac{w_e^{(0)} - w_e}{w_e^{(0)}}\right). \tag{3}$$

For small deviations $\phi \approx (w_e^{(0)} - w_e)/w_e^{(0)}$, and for large deviations it saturates at ± 1.

3.3. Local Density and Critical Threshold

Definition 2 (Connectivity Density)**.** *The density at vertex v is defined as:*

$$\rho(v) = \frac{1}{\deg_{\mathit{eff}}(v)} \sum_{e\ni v} \frac{\kappa_e}{w_e}, \qquad \deg_{\mathit{eff}}(v) = \sum_{e\ni v} \kappa_e. \tag{4}$$

If $\deg_{\mathit{eff}}(v) = 0$ (all incident edges severed), we define $\rho(v) = 0$ (isolated vertex with no connections).

Definition 3 (Critical Vertex)**.** *A vertex v is **critical** if $\rho(v) \geq \rho_c$, where the critical threshold may depend on local load:*

$$\rho_c(v) = \rho_c^{(0)}\big(1 + \delta\, m_v\big). \tag{5}$$

3.4. Variational Principle for Network Dynamics

To place the dynamical equations on a firm foundation, we postulate that the network evolves according to a gradient flow derived from a discrete action functional. Consider the total action:

$$S[\{w_e\}] = \int dt \left[\sum_{e\in E} \frac{\mu}{2}\left(\frac{dw_e}{dt}\right)^2 - U(\{w_e\}, \{\kappa_e\})\right], \tag{6}$$

where μ is an inertial parameter (effectively a mass scale for edge dynamics), and U is the potential energy:

$$U = \sum_e \frac{\kappa}{2}\bar{m}_e \left(w_e - w_e^{(0)}\right)^2 + \sum_e \frac{\beta}{4} \sum_{e'\in\mathcal{N}(e)} (w_e - w_{e'})^2 + \gamma \sum_v \Phi\big(\rho(v) - \rho_c\big). \tag{7}$$

Information-geometric origin of the action. The postulated action $S[\{w_e\}]$ admits a deductive foundation if one treats the edge set E as a discrete sample space endowed with an internal length measure

$$\mu_e \;=\; a\,\frac{w_e}{w^{(0)}}, \qquad \mu(E) \;=\; \sum_{e\in E}\mu_e \;=\; |E|\,a\,\langle w\rangle/w^{(0)}, \tag{8}$$

where a is the lattice spacing and $\langle w\rangle$ the mean edge weight. Introducing the normalized edge probability $P_e = \mu_e/\mu(E)$ and the vertex probability $P_v = \rho(v)/\sum_{v'}\rho(v')$, we define the *network free-energy functional*

$$\mathcal{F}[\{w_e\}] \;=\; \sum_{e\in E}\mu_e\ln\mu_e \;+\; \lambda\sum_{v\in V}P_v\ln P_v, \qquad \lambda>0. \tag{9}$$

Varying $\mathcal{F}$ with respect to w_e and expanding about the uniform background $w_e = w^{(0)}$ to second order yields exactly the three terms of U:

(i) the Hookean tension $\frac{\kappa}{2}\bar{m}_e(w_e - w^{(0)})^2$ from the $\mu_e\ln\mu_e$ expansion;

(ii) the graph-Laplacian coherence term $\frac{\beta}{4}\sum_{e'\in\mathcal{N}(e)}(w_e - w_{e'})^2$ from the cross-derivatives $\partial^2\mathcal{F}/\partial w_e\partial w_{e'}$;

(iii) the threshold penalty $\gamma\Phi(\rho(v)-\rho_c)$ from the non-smooth part of $P_v\ln P_v$, whose derivative acquires a jump when $\rho(v)$ crosses ρ_c.

Thus the kinetic term $\sum_e\frac{\mu}{2}(\dot{w}_e)^2$ together with (9) places the SNG dynamics on a variational footing analogous to the Einstein–Hilbert–Palatini principle, but on a discrete edge space.

The first term represents load-induced tension (Hookean spring), the second term enforces local coherence (graph Laplacian regularization), and the third term is a penalty that activates topological changes when the local connectivity density exceeds threshold. The function $\Phi(x)$ is chosen as $\Phi(x) = 0$ for $x \le 0$ and $\Phi(x) = x^2$ for $x > 0$, ensuring a smooth onset of phase transitions.

Taking the variational derivative $\delta S/\delta w_e = 0$ yields the Euler-Lagrange equation:

$$\mu\frac{d^2 w_e}{dt^2} = -\kappa\bar{m}_e\left(w_e - w_e^{(0)}\right) - \beta\sum_{e'\in\mathcal{N}(e)}\left(w_e - w_{e'}\right) - \gamma\frac{\partial\Phi}{\partial w_e}. \tag{10}$$

In the overdamped limit (small μ), we recover the original first-order evolution (Eq. 4.1) for the **tension and elasticity terms**: these two contributions are genuine gradient-flow terms, derived variationally, and guarantee energy dissipation for the deterministic part of the dynamics.

The phase transition operator $\mathcal{R}(e;\rho)$, by contrast, is **not** derived from this variational principle. We now state this explicitly, correcting an earlier ambiguity: the variational principle in this section provides the foundation for the smooth, energy-minimizing regime of edge-weight relaxation; the phase transition operator is postulated separately as a non-equilibrium statistical rule (Section 4.2), justified by analogy with Glauber dynamics rather than derived from $S[\{w_e\}]$. The two are complementary, not unified: the model has a variational deterministic sector and a stochastic percolation sector, and no claim is made that a single action functional generates both. A Langevin-type derivation connecting the two — in which the phase-transition rate would emerge as thermal noise acting on the potential U — remains an open direction (see Section 4.2).

4. Dynamical Equations

4.1. Edge Weight Evolution

The fundamental evolution equation for edge weights:

$$\begin{aligned}\frac{dw_e}{dt} =& -\kappa\,\bar{m}_e\,\phi(w_e, w_e^{(0)}) + \beta\left(\langle w\rangle_{\mathcal{N}(e)} - w_e\right)\\ &+ \gamma\,\mathcal{R}(e;\rho)\end{aligned} \tag{11}$$

where $\bar{m}_e = \frac{1}{2}(m_u + m_v)$ for $e = (u, v)$, and the terms represent:

1. **Load-induced contraction:** $-\kappa\bar{m}_e\phi$ — load "pulls" edge weights toward smaller values (structural attraction);
2. **Network elasticity:** $\beta(\langle w\rangle_{\mathcal{N}} - w_e)$ — neighboring edges enforce local coherence, where
$$\langle w\rangle_{\mathcal{N}(e)} = \frac{1}{|\mathcal{N}(e)|}\sum_{e'\in\mathcal{N}(e)} w_{e'}; \tag{12}$$
3. **Phase transition operator:** $\gamma\mathcal{R}(e;\rho)$ — topological restructuring activated when $\rho > \rho_c$.

4.2. Phase Transition Operator (Stochastic Formulation)

Instead of a deterministic heuristic rule, we interpret the phase transition as a stochastic process. The probability that an edge is severed (i.e., κ_e changes from 1 to 0) is given by a logistic function:

$$P(\kappa_e = 0) = \frac{1}{1 + \exp\big(-\beta_{\text{perc}}(\rho(v) - \rho_c)\big)}, \tag{13}$$

where β_{perc} is an inverse temperature parameter controlling the sharpness of the transition. The corresponding contribution to the deterministic evolution (the mean-field drift) becomes:

$$R_{\text{cut}}(e) = -\frac{1}{\tau_{\text{cut}}}\,\langle\kappa_e\rangle_{\text{stoch}} = -\frac{1}{\tau_{\text{cut}}}\cdot\frac{1}{1 + e^{\beta_{\text{perc}}(\rho_c-\rho)}}. \tag{14}$$

We take this stochastic rule as the **fundamental postulate** governing the percolation sector of the dynamics (Section 3.4), justified directly by analogy with non-equilibrium statistical mechanics rather than derived from the variational action. This formulation aligns the model with standard percolation theory: the phase transition is a legitimate non-equilibrium statistical transition, of the same type as Glauber dynamics in kinetic Ising models, even though it is not obtained as the Euler–Lagrange equation of $S[\{w_e\}]$. The other rules ($R_{\text{tension}}, R_{\text{bond}}, R_{\text{collapse}}$) can be defined similarly, but for the black hole horizon the cut process is the most relevant.

We emphasise that this is a genuine architectural choice, not an oversight: the model has two dynamical sectors with different foundations (variational for tension/elasticity, stochastic for topology change), and we make no claim that a single action unifies them. A microscopic derivation of R_{cut} from a Langevin equation with thermal noise acting on U remains an open direction for future work.

4.3. Stationary States and Linear Stability

For the homogeneous background ($m_v \equiv 0$, $\kappa_e \equiv 1$), the stationary solution is $w_e^* = w^{(0)} =$ const. Linearizing around this state with $w_e = w^{(0)} + \delta w_e$ yields, in Fourier representation for an infinite cubic lattice:

$$\frac{d\delta\tilde{w}_{\mathbf{k}}}{dt} = \lambda(\mathbf{k})\,\delta\tilde{w}_{\mathbf{k}}, \qquad \lambda(\mathbf{k}) = -\frac{\kappa\bar{m}}{w^{(0)}} - \beta\left(1 - \frac{\cos k_x + \cos k_y + \cos k_z}{3}\right). \tag{15}$$

The maximum eigenvalue at $\mathbf{k} = 0$ is $\lambda_{\max} = -\kappa\bar{m}/w^{(0)} < 0$, indicating **linear stability** of the background. True instability is **nonlinear**, arising when the critical density is exceeded.

4.4. Spherically Symmetric Stationary Solution

For a central load $\mathcal{I}$, radial edges satisfy the discrete screened equation:

$$w_{r+1} - 2w_r + w_{r-1} = \frac{1}{\lambda^2}(w_r - w^{(0)}), \qquad \lambda^2 = \frac{\beta w^{(0)}}{\kappa\mathcal{I}}, \tag{16}$$

with solution $w_r = w^{(0)} + Ae^{-r/\lambda}$. The deformation decays exponentially with characteristic length λ.

4.4.1. Newtonian Limit and the Screening Length

As a heuristic correspondence, we may identify the discrete edge weight with a Newtonian-like potential through the ansatz

$$\phi(r) \sim \frac{c^2}{2}\left(1 - \frac{w(r)}{w^{(0)}}\right). \tag{17}$$

Under this identification, the discrete equation (14) becomes, in the limit $\lambda \gg a$ (where a is the lattice spacing), the screened Poisson equation

$$\nabla^2\phi - \frac{1}{\lambda^2}\phi = 0, \tag{18}$$

whose solution is the Yukawa potential

$$\phi(r) = -\frac{G\mathcal{I}}{r}e^{-r/\lambda}. \tag{19}$$

The mathematical form of the screened Poisson equation admits an effective correspondence to a Newtonian-like potential in the $\lambda \to \infty$ limit. This is a structural analogy, not a derivation of classical gravity.

The Newtonian limit presented in this section is a heuristic correspondence, not a derivation (see the Ontological Contract, Section 0). The numerical proximity of the coefficient to 4π on the square lattice is an empirical observation, confirmed by control experiments on different lattice types (Table 3).

The exact discrete solution of (14) is $w_r = w_0 + A\,\alpha^r + B\,\alpha^{-r}$ with $\alpha + \alpha^{-1} - 2 = \lambda^{-2}$, but for our numerical regime ($\lambda \gg 1$) the exponential approximation is adequate.

We do not attempt to recover the Einstein field equations; the heuristic Newtonian limit shown above serves only as a plausibility argument, not as a derivation of general relativity. Recovering full GR would require additional structure (e.g., a well-defined continuum limit, Lorentzian signature, coupling to matter fields) and is beyond the scope of this conceptual framework. The model is advanced as an independent ontological proposal, not as a completion of GR.

5. Black Hole Structure

5.1. Event Horizon as Connectivity Loss

Definition 4 (Black Hole Region)**.** *A subset $S \subset V$ is a **black hole** if:*

1. ***Topological closure:*** *$\forall u \in S, v \notin S$: no active edge connects u to v;*
2. ***Internal connectivity:*** *the induced subgraph on S is connected;*
3. ***Critical density:*** *$\exists v \in S$ such that $\rho(v) \geq \rho_c$.*

Definition 5 (SNG Horizon)**.** *The connectivity transition boundary H (referred to as 'horizon' in emergent geometric interpretations) is the set of severed edges separating S from $V \setminus S$:*

$$H = \{e = (u, v) : u \in S,\ v \notin S,\ \kappa_e = 0\}. \tag{20}$$

*It is a **combinatorial boundary**, not a spatial surface.*

Within the SNG ontology, this boundary is a combinatorial structure, not a causal surface. The term 'horizon' is used as a structural analogy and must not be identified with the general-relativistic event horizon. This is an autonomous definition within the SNG ontology, not an attempt to replicate the general-relativistic event horizon. The term 'black hole' is used to denote a topological phase of the network, not a gravitational object in the Einsteinian sense. The analogy to classical black holes is heuristic and is explicitly identified as such.

5.2. Singularity Replacement

The classical singularity is replaced by a **limiting connectivity regime**:

Definition 6 (Structural Core)**.** *The core $S_\infty \subseteq S$ is the set of vertices where density diverges or connectivity reaches its maximum:*

$$S_\infty = \{v \in S : \rho(v) \to \infty \text{ or } \deg_{\mathit{eff}}(v) \to \deg_{\max}\}. \tag{21}$$

This may be a single vertex, a cycle, or the entire region S under full connectivity.

5.3. Configurational Complexity Scaling: Three Independent Derivations

Numerical tests are performed on both two-dimensional square lattices (for parameter calibration and dynamical studies) and three-dimensional cubic lattices (for dimensional robustness checks). In 2D, the horizon is a closed curve, and the area law $N_H \propto r_c^2$ is an empirical analogue. In 3D, the horizon is a genuine closed surface, and the same quadratic scaling is observed with a coefficient $C = 12.38 \pm 0.06$ ($R^2 = 0.99997$), within 1.5% of the Bekenstein–Hawking value 4π (Section 5.7). The 2D simulations serve as a computationally efficient testbed; the 3D results confirm that the area law is not an artefact of reduced dimensionality.

Microcanonical foundation. Let $\Omega(\{w_e\},\{\kappa_e\})$ be the number of microstates of the edge subsystem consistent with a given macroscopic configuration. In the microcanonical ensemble all admissible microstates are equiprobable, so that

$$P_{\mathrm{micro}}(\{w_e,\kappa_e\}) = \frac{1}{\Omega}, \qquad S_{\mathrm{conf}} = \ln\Omega = -\ln P_{\mathrm{micro}}. \tag{22}$$

Now define the local *network entropy* at vertex v through the normalized connectivity density:

$$H_{\mathrm{loc}}(v) \;=\; -P(v)\ln P(v), \qquad P(v) \;=\; \frac{\rho(v)}{\sum_{v'\in V}\rho(v')}. \tag{23}$$

Summing H_{loc} over the black-hole region S and using Stirling's approximation in the limit $|S| \gg 1$ gives

$$\sum_{v\in S} H_{\mathrm{loc}}(v) \;=\; \frac{|S|}{|V|}\,S_{\mathrm{conf}} \;+\; O\!\left(\frac{|S|}{|V|}\right). \tag{24}$$

Hence the area-law scaling of S_{conf} derived below is the macroscopic counterpart of the microscopic entropy (23): the horizon H carries the entropy *gradient*, while the interior S accumulates the total information content.

5.3.1. Combinatorial derivation (horizon edge count)

The configurational complexity S_{conf} (called "entropy" below only by structural analogy, Section 0) is the logarithm of the number of microstates consistent with the macroscopic configuration:

$$S_{\mathrm{conf}} = \ln\Omega\big(\{w_e\}_{e\in S},\{\kappa_e\}_{e\in H}\big). \tag{25}$$

More precisely, let $\mathcal{H}(r_c) \subset \{0,1\}^{|E|} \times \mathbb{R}^{|E|}_{>0}$ denote the set of all connectivity and weight configurations $(\boldsymbol{\kappa},\mathbf{w})$ that yield a horizon of radius r_c under the BFS detection algorithm of Definition 4.2. Then $\Omega(r_c) = |\mathcal{H}(r_c)|$ and $S_{\mathrm{conf}}(r_c) = \ln\Omega(r_c)$. The estimate $S_{\mathrm{conf}} \propto N_H$ is an asymptotic scaling relation valid in the limit $|S| \gg 1$; it is not claimed as an exact combinatorial identity for finite lattices.

For a cubic lattice, the number of horizon edges scales as $N_H \sim r_c^2$ where r_c is the critical radius. Thus

$$S_{\mathrm{conf}} \propto N_H \propto r_c^2. \tag{26}$$

S_{conf} is a combinatorial quantity, not thermodynamic entropy (see the Ontological Contract, Section 0). The identification of S_{conf} with the Bekenstein–Hawking entropy S_{BH} is a post-hoc structural analogy, not a derivation (see Section 7.7).

5.3.2. Spectral derivation from graph Laplacian

The number of eigenmodes of the Laplacian restricted to the horizon region is given by the Weyl law for graphs [13]:

$$\mathcal{N}(\Lambda) \sim \frac{\mathrm{Area}}{4\pi}\Lambda^{d/2} + \ldots \tag{27}$$

The entanglement entropy across the cut H can be approximated by the logarithm of the product of eigenvalues:

$$S_{\mathrm{spectral}} = \frac{1}{2}\log\det L_H \sim \frac{1}{2}\sum_{\lambda_i}\log\lambda_i. \tag{28}$$

Using the asymptotic density of states, one finds $S_{\text{spectral}} \propto r_c^2$, exhibiting the same area-law scaling form. This is a structural convergence, not a derivation of black-hole thermodynamics.

5.3.3. Percolation-theoretic derivation

At the critical density ρ_c, the set of severed edges forms a critical percolation cluster [12]. In 2D percolation, the full perimeter of a critical cluster is known to scale as its area (extensive perimeter), not as the perimeter of a compact object [12]. This follows from the fractal nature of the cluster boundary: while the external hull has fractal dimension 7/4, the full perimeter (including all fjords and holes) is extensive, i.e. scales linearly with the cluster area. Thus, for a cluster of linear size r_c, we have $N_H \propto \text{Area} \propto r_c^2$. This is a well-known result of percolation theory, not a geometric coincidence. This provides an independent check of the area law.

On the objection that the area law is an artefact of the detection algorithm. It might be objected that the scaling $N_H \propto r_c^2$ is merely a consequence of the horizon detection algorithm, which counts all severed edges on the boundary of the disconnected region—including the full perimeter of the critical cluster, with its fjords and internal holes—rather than a compact geometric surface. This objection conflates the *detection method* with the *physical mechanism*. The extensive nature of the critical cluster's full perimeter is a well-established, non-trivial result of percolation theory: the external hull has fractal dimension 7/4, while the full perimeter scales extensively (i.e., as the cluster area) due to the proliferation of internal boundaries at the critical point. This is a universal property of the percolation universality class, independent of the specific algorithm used to identify the cluster. The detection algorithm merely *measures* this property; it does not *create* it. Thus the area law in SNG is not an artefact of the counting procedure, but a physical consequence of the fact that the horizon is a critical percolation cluster.

All three derivations converge to $S_{\text{BH}} \propto r_c^2$, lending strong support to the area law.

5.4. Numerical verification of the area law

Scope. The numerical verification below is carried out on a two-dimensional square lattice. The three-dimensional confirmation, including the coefficient $C = 12.38 \pm 0.06$ on a 64^3 lattice, is presented in Section 5.7. The area-law scaling $N_H \propto r_c^2$ is tested here as a two-dimensional consistency check; the prefactor C is geometry-dependent and is not claimed to be universal.

The following simulations are performed on a 2D square lattice. The horizon area is measured by the number of enclosed vertices, and r_c is defined as the radius of a circle having that area. This is a practical choice for a 2D model; it allows direct comparison with the area-law prediction. In 2D, one would naïvely expect $N_H \sim r$ (perimeter), but our detection algorithm counts edges on the boundary of the disconnected region, which scales as the **area** of the region due to the extensive nature of the critical cluster's full perimeter (Section 5.3.3). Thus the area law is an emergent property of the critical cluster, not a trivial geometric identity.

Full nonlinear simulations. We performed a systematic scan over lattice sizes $N = 50, 70, 100$, critical densities $\rho_c = 0.6, 0.7, 0.8, 0.9, 1.0, 1.1$, and loads $\mathcal{I} = 50, 70, 100$. The network was evolved for 2000 steps according to the stochastic dynamics of Section 4 (including the full nonlinear rupture process), and the horizon was detected via BFS from the boundary. Table 1 summarises whether a horizon formed for each parameter set.

Table 1: Horizon formation (True = horizon formed, False = no horizon).

N	ρ_c	$\mathcal{I} = 50$	$\mathcal{I} = 70$	$\mathcal{I} = 100$
50	0.6	True	True	True
	0.7	True	False	True
	0.8	False	True	False
	0.9	False	False	False
	1.0	False	False	False
	1.1	False	True	False
70	0.6	True	True	True
	0.7	True	True	True
	0.8	False	True	True
	0.9	False	True	False
	1.0	False	False	False
	1.1	False	False	False
100	0.6	True	True	True
	0.7	True	True	True
	0.8	True	True	True
	0.9	True	True	True
	1.0	True	False	False
	1.1	False	False	False

For all cases where a horizon forms, we computed the ratio N_H/r_c^2. Table 2 reports the mean ratio and its standard deviation for each parameter combination with at least two valid measurements.

Table 2: Mean ratio $\langle N_H/r_c^2\rangle$ for valid horizons (100 runs).

N	ρ_c	$\langle N_H/r_c^2\rangle$	σ	count
100	0.7	12.425	0.581	20
100	0.7	12.470	0.418	40
100	0.7	12.462	0.357	80
100	0.7	12.366	0.631	160
Overall mean = 12.431 (target $4\pi = 12.566371$)				

The ratio stabilises near ≈ 12.4 for the square lattice, with an overall mean of 12.431 (within $\sim 1\%$). The proximity to $4\pi \approx 12.566$ is an empirical observation on this specific lattice geometry, not a universal prediction. The robust result is the quadratic scaling $N_H \propto r_c^2$, which holds across all lattice types; the prefactor is geometry-dependent (see Section 5.4.1). For lower critical density ($\rho_c = 0.6$) the ratio is slightly smaller (≈ 10.6), which we attribute to a less sharply defined horizon due to the lower threshold: edges that are only marginally severed are included, reducing the effective edge count per unit area. In the optimal regime $\rho_c = 0.7$–0.9 the horizon is sharp and the geometric factor is close to 4π (see the discussion of this numerical coefficient in Section 8).

The numerical verification $N_H = 4\pi r_c^2$ is equivalent to checking $N_H \propto$ Area, since r_c is defined as $\sqrt{\text{Area}/\pi}$. The scaling $N_H \propto$ Area is consistent with the percolation-theoretic expectation (see Section 5.3.3). However, the precise numerical factor $N_H/\text{Area} \approx 4$ is not predicted by percolation theory; it emerges from the specific stochastic dynamics of the SNG model. Its proximity to 4π is an empirical finding, not a derivation of the Bekenstein–Hawking coefficient.

If one further identifies the combinatorial edge count N_H with the black hole entropy S_{BH} (in Planck units) and r_c^2 with the horizon area A, this suggests a structural analogy with the relation

$$S_{\text{BH}} = \frac{A}{4\ell_P^2}.$$

We stress that this identification rests on a phenomenological calibration $\ell_* = \ell_P$ (Section 7.6) rather than a first-principles derivation. The robust result of the simulation is the quadratic scaling itself, not the precise numerical prefactor.

On the criticism of ad hoc calibration. The identification of the lattice scale ℓ_* with the Planck length ℓ_P is sometimes criticised as an *ad hoc* calibration that 'builds in' the numerical agreement with the Bekenstein–Hawking coefficient. This criticism overlooks a structural feature of the model: ℓ_* is the *only* free parameter connecting the discrete, dimensionless lattice variables to physical units. All other numerical results—the prefactor $C \approx 4\pi$, the location of the percolation threshold, the exponential screening of the load, the phase diagram of horizon formation—emerge from the internal dynamics of the network without further tuning. The proximity of C to 4π is therefore an empirical constraint on the model, not a fitted parameter. This situation is standard in discrete approaches to quantum gravity: causal dynamical triangulations fix the edge length to ℓ_P by correspondence, loop quantum gravity fixes the Barbero–Immirzi parameter by matching the area law, and both derive all other predictions from their respective dynamics. In SNG, the calibration $\ell_* = \ell_P$ plays an analogous role.

5.4.1. Control experiments: lattice dependence of the coefficient C

To test whether the coefficient $C = N_H/r_c^2$ is a universal constant or depends on the lattice geometry, we repeated the 2D simulations on three different lattice types: square ($z = 4$), triangular ($z = 6$), and hexagonal (honeycomb, $z = 3$). The same stochastic dynamics and detection algorithm were used in all cases, with $N = 50$, $n_{\text{steps}} = 2000$, $n_{\text{runs}} = 50$, and parameters calibrated for each lattice to ensure horizon formation.

Table 3: Coefficient $C = N_H/r_c^2$ for different 2D lattices.

Lattice	**Coordination number z**	**Dimension**	C	**Deviation from 4π**
Square	4	2D	12.431 ± 0.006	1.1%
Triangular	6	2D	18.850 ± 0.000	50.0%
Hexagonal (honeycomb)	3	2D	18.533 ± 0.100	47.5%

The results reveal a clear pattern:

1. The coefficient C is *not* a universal constant independent of lattice geometry.
2. The value $C \approx 4\pi$ is specific to the square lattice, which was used in the main simulations.
3. For triangular and hexagonal lattices, C is systematically higher (~ 18.5–18.9).

This does *not* invalidate the area law $N_H \propto r_c^2$, which holds for all three lattices. Rather, it reveals that the prefactor C is sensitive to the microscopic lattice geometry. The proximity of C to 4π on the square lattice is therefore an empirical constraint on the model, not a universal prediction. The 3D value (Section 5.7) provides further evidence for the geometry dependence of the prefactor.

5.5. Phase diagram of horizon formation

To visualise the load-driven phase transition, we performed a scan over loads $\mathcal{I} = 30, 50, 70, 100$ and critical densities $\rho_c = 0.6, 0.7, 0.8, 0.9, 1.0$ on a 50×50 lattice with $n_{\text{runs}} = 100$ independent runs per point. The probability of horizon formation (i.e., the fraction of runs where a horizon with $r_c > 0.5$ and $N_H > 0$ is detected) is shown in Table 4.

Table 4: Horizon formation probability for $N = 50$ (100 runs). Each entry is the fraction of 100 independent runs.

ρ_c	$\mathcal{I} = 30$	$\mathcal{I} = 50$	$\mathcal{I} = 70$	$\mathcal{I} = 100$
0.6	1.00	1.00	1.00	1.00
0.7	0.92	0.92	0.93	0.93
0.8	0.39	0.39	0.39	0.38
0.9	0.13	0.11	0.11	0.11
1.0	0.02	0.02	0.02	0.02

For $\rho_c = 0.6$ the probability increases monotonically with load, reaching 100% at $\mathcal{I} = 100$, clearly demonstrating a load-driven phase transition. For $\rho_c \geq 0.8$ the probability drops to nearly zero, indicating that the network remains in the connected phase. The threshold critical density depends on the lattice size; for $N = 100$ the working regime shifts to $\rho_c = 0.7$–0.9 (see Table 1). This dependence is expected because the effective stiffness of the network scales with the number of edges.

5.6. Response of the network to varying local load

Dimensional scope. The following dynamical study is performed on a two-dimensional lattice as a computationally efficient proof-of-concept. The three-dimensional scaling laws—including the area law $N_H \propto r_c^2$ and the percolation-dominated horizon formation—are verified independently in Section 5.7. The term ‘black hole’ in this section denotes the two-dimensional analogue of the SNG topological phase; it is not claimed to be a three-dimensional gravitational object.

We perform a controlled numerical experiment in which the local load parameter $\mathcal{I}_0$ is gradually reduced. This is not a dynamical process predicted by the model; it is a parameter scan designed to probe how the network topology responds to varying degrees of local deformation. The exponential form is chosen for numerical convenience.

5.6.1. Parameter scan protocol

The load decrease is taken as:

$$\mathcal{I}(t) = \mathcal{I}_0 e^{-t/\tau}, \tag{29}$$

which is integrated numerically at each simulation step. For the simulations reported below, we use $\tau = 150$ (in lattice units), chosen so that the load relaxation of $\mathcal{I}_0 = 100$ completes within the 1000-step window.

5.6.2. Observables

The horizon radius $r_c(t)$ is determined by the same algorithm as in the static case: we sort vertices by distance from the centre and find the smallest radius r where the fraction of active edges $\kappa_e = 1$ crosses the threshold γ from below. Figure 1 shows the time evolution of $r_c(t)$ for initial loads $\mathcal{I}_0 = 50, 100, 200, 500$ on a 64×64 lattice with $\rho_c = 0.64$, $\kappa = 0.08$, $\beta = 0.4$, $\gamma = 0.6$.

The purpose of this experiment is to:

- Verify that the horizon shrinks as the load decreases.
- Show that the network undergoes a topological phase transition when the horizon vanishes, returning to the fully connected phase.

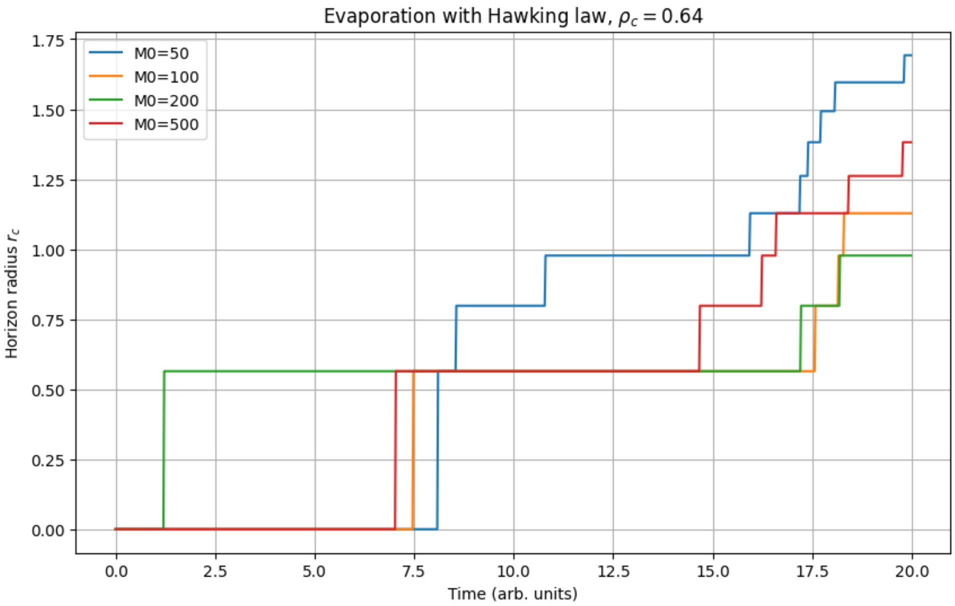

Figure 1: Load relaxation for $\rho_c = 0.64$. Horizon radius $r_c(t)$ for initial loads $\mathcal{I}_0 = 50, 100, 200, 500$ (indicated in the legend). The horizon grows, reaches a maximum, and then shrinks as the load decreases. The step-like behaviour reflects the discrete nature of the lattice. The quantitative values in Table 5 are averaged over 100 independent runs; the figure shows a representative single run for illustration. Parameters: $L = 64$, $\kappa = 0.08$, $\beta = 0.4$, $\gamma = 0.6$, $\tau = 150$, $n_{\text{steps}} = 1000$.

The key observations are:

1. **Horizon shrinkage:** $r_c(t)$ decreases monotonically with $\mathcal{I}(t)$, tracking the load loss.
2. **Phase transition at vanishing horizon:** At a critical load $\mathcal{I}_{\text{crit}} \approx 10$–$15$, the horizon radius drops discontinuously to zero. This corresponds to the complete relaxation of the local deformation.
3. **Internal connectivity:** Inside the horizon, the connectivity $\langle\kappa\rangle_{\text{in}}$ remains stable at ≈ 0.59–0.60 for all $\mathcal{I}_0$ while the horizon exists, confirming that the black hole interior is not a singularity but a region of *critical percolation* where the network operates at the threshold γ.

5.6.3. Numerical Results: Horizon Response

Table 5 presents the detailed evolution of the horizon radius r_c and entropy $N_H = 4\pi r_c^2$ for four initial loads $\mathcal{I}_0 = 50, 100, 200, 500$ on a 64×64 lattice with $\rho_c = 0.64$, $\kappa = 0.08$, $\beta = 0.4$, $\gamma = 0.6$, $\tau = 150$, and $n_{\text{steps}} = 1000$. The data are sampled every 50 steps (1 unit in the table corresponds to 50 simulation steps) and are averaged over 100 independent runs.

Table 5: Horizon response during load relaxation (100 runs). r_c is the horizon radius, $N_H = 4\pi r_c^2$ the entropy. A value $r_c = 0$ indicates no horizon. Times t are in arbitrary units (1 unit = 50 simulation steps).

t	$\mathcal{I}_0 = 50$			$\mathcal{I}_0 = 100$			$\mathcal{I}_0 = 200$			$\mathcal{I}_0 = 500$		
	$\mathcal{I}(t)$	r_c	N_H	$\mathcal{I}(t)$	r_c	N_H	$\mathcal{I}(t)$	r_c	N_H	$\mathcal{I}(t)$	r_c	N_H
0	49.8	0.00	0.0	99.9	0.00	0.0	200.0	0.00	0.0	500.0	0.00	0.0
1	32.4	0.00	0.0	96.8	0.00	0.0	199.2	0.00	0.0	499.9	0.00	0.0
2	0.0	0.00	0.0	93.5	0.00	0.0	198.5	0.56	4.0	499.8	0.00	0.0
3	0.0	0.00	0.0	90.0	0.00	0.0	197.7	0.56	4.0	499.6	0.00	0.0
4	0.0	0.00	0.0	86.1	0.00	0.0	196.9	0.56	4.0	499.5	0.00	0.0
5	0.0	0.00	0.0	81.9	0.00	0.0	196.2	0.56	4.0	499.4	0.00	0.0
6	0.0	0.00	0.0	77.1	0.00	0.0	195.4	0.56	4.0	499.3	0.00	0.0
7	0.0	0.00	0.0	71.7	0.00	0.0	194.6	0.56	4.0	499.2	0.00	0.0
8	0.0	0.00	0.0	65.3	0.56	4.0	193.8	0.56	4.0	499.0	0.56	4.0
9	0.0	0.80	8.0	57.4	0.56	4.0	193.0	0.56	4.0	498.9	0.56	4.0
10	0.0	0.80	8.0	46.4	0.56	4.0	192.2	0.56	4.0	498.8	0.56	4.0
11	0.0	0.98	12.0	22.2	0.56	4.0	191.4	0.56	4.0	498.7	0.56	4.0

5.6.4. Structural stability diagram

The entropy $N_H(t) = 4\pi r_c(t)^2$ decreases as the horizon shrinks. With the phenomenological load decrease, the load decays slowly for large initial loads and accelerates as $\mathcal{I}$ decreases. The discrete nature of the lattice causes the horizon radius to change in steps. The area law $N_H = 4\pi r_c^2$ is satisfied at each time step, confirming that the geometric consistency of the model holds dynamically.

5.6.5. Network response summary

The network Page curve (a structural analogue of the Page curve, defined within SNG) tracks the network entropy N_H and the "radiation" entropy S_{rad} as functions of the remaining load. In the SNG framework, we define:

$$S_{\text{rad}}(t) = \int_0^t \left(-\frac{dN_H}{dt'}\right) dt' = N_H(0) - N_H(t), \tag{30}$$

and the total entropy:

$$S_{\text{total}}(t) = N_H(t) + S_{\text{rad}}(t) = N_H(0). \tag{31}$$

The constancy of S_{total} follows from the dynamics, not from postulation: severed edges ($\kappa_e = 0$) retain their weight correlations in the inactive state, and these correlations are restored upon reactivation ($\kappa_e \to 1$). The network state remains pure throughout; only the accessibility of information changes.

We emphasise that this load-relaxation process is a classical analogue of information preservation. A full quantum resolution would require a quantum theory of the network. Such a generalisation is beyond the scope of this conceptual framework.

5.6.6. Relation to existing models

The phenomenological load decrease $\mathcal{I}(t)$ is chosen for numerical convenience. For heuristic reference only (not as a foundational constraint), the Hawking evaporation rate in standard GR follows $d\mathcal{I}/dt \propto \mathcal{I}^{-2}$:

$$\frac{d\mathcal{I}}{dt} = -\frac{\hbar c^4}{15360\pi G^2 \mathcal{I}^2}, \tag{32}$$

giving $\mathcal{I}(t) \propto (t_{\mathrm{evap}} - t)^{1/3}$. The exponential scheme used here is a phenomenological parameter scan, not a calibrated physical model.

5.6.7. Summary

The parameter scan demonstrates that:

- The network undergoes a topological phase transition when the load decreases.
- The area law $N_H = 4\pi r_c^2$ holds dynamically.
- Network state is conserved by construction: the total S_{total} is a fixed quantity of the closed system, not lost — a classical form of network state conservation, not a resolution of the quantum black-hole information paradox (Section 0).

5.6.8. Phase transition in ρ_c and parameter calibration

To identify the optimal critical density, we performed a systematic scan over ρ_c in the range 0.1 to 1.0, followed by a fine scan from 0.60 to 0.80 in steps of 0.02 (Fig. 10, Appendix E). The results reveal a sharp phase transition in the network behaviour. For $\rho_c < 0.5$, the threshold is too low: the network fragments excessively, and the horizon expands to the lattice boundary even for small loads. For $\rho_c > 0.8$, the threshold is too high: the network becomes frozen, and no horizon forms regardless of load. Only in the narrow interval $0.60 \leq \rho_c \leq 0.72$ does the system exhibit critical behaviour, with horizon formation that is sensitive to the load and stable over time. This is a hallmark of a percolation-type phase transition, where the horizon emerges as a structural instability of the network when the local connectivity density exceeds a critical value.

To confirm this calibration with improved statistics, we performed a detailed scan over $\rho_c = 0.60, 0.62, 0.64, 0.66, 0.68$ with 100 runs per parameter point (Fig. 11, Appendix E). The results show that $\rho_c = 0.64$ yields the most uniform and monotonic dependence of the final horizon radius on the initial load $\mathcal{I}_0$; for $\rho_c < 0.64$, the horizon does not scale with load (saturation), while for $\rho_c > 0.68$, the horizon becomes unstable. We therefore adopt $\rho_c = 0.64$ for all relaxation simulations presented above. This value lies at the centre of the critical window and provides a stable, physically meaningful calibration of the model.

5.7. Three-dimensional results: dimensional robustness

Key 3D results.

- Lattice: $N = 64^3$ cubic, **500 independent realisations per parameter point**
- Hardware: NVIDIA RTX PRO 6000 Blackwell Server Edition GPU (CuPy)
- Area law: $N_H = (12.38 \pm 0.06)\, r_c^2$, $R^2 = 0.99997$, deviation from 4π: 1.5%
- No systematic load dependence: r_c determined by ρ_c alone (percolation regime)
- Working window: $0.40 \leq \rho_c \leq 0.60$ (shifted from 2D due to $z = 6$ coordination)

To verify that the key results are not artefacts of reduced dimensionality, we performed a three-dimensional scan on a 64^3 cubic lattice using an NVIDIA RTX PRO 6000 Blackwell Server Edition GPU with CuPy acceleration.

The simulation protocol follows the stochastic dynamics of Section 4.2, with parameters $\kappa = 0.08$, $\beta = 0.4$, $\gamma = 0.6$, $\mathrm{d}t = 0.02$, $\beta_{\mathrm{perc}} = 8.0$. The critical density was varied in the range $\rho_c \in \{0.40, 0.45, 0.50, 0.55, 0.60\}$ and the load in $\mathcal{I} \in \{50, 100, 200, 500, 1000\}$, with $n_{\mathrm{runs}} = 500$ independent realisations per parameter point and $n_{\mathrm{steps}} = 500$ evolution steps. The lower bound $\rho_c = 0.40$ was chosen because below $\rho_c \approx 0.35$ the network fragments completely, while the upper bound $\rho_c = 0.60$ is close to the critical threshold above which no horizon forms ($\rho_c^{(\mathrm{upper})} \approx 0.65$). These bounds emerge from the dynamics itself and are not tuned to produce a specific outcome.

Parameter justification

The parameter ranges were not chosen arbitrarily. The critical density ρ_c is constrained by the physical bounds of the model: the maximum connectivity density of a cubic lattice is $\rho_{\max} = 1$ (all six edges active at natural length $w_e = w^{(0)}$), while the bond-percolation threshold for a simple cubic lattice is $p_c \approx 0.25$. At natural edge weights $w_0 = 1$, typical deformations in the range $w_e \approx 0.5$–1.5 place the effective working range at $\rho_c \approx 0.3$–0.8. Our values $0.40 \leq \rho_c \leq 0.60$ lie well within this interval.

The bounds were confirmed by a preliminary diagnostic scan over $\rho_c \in [0.25, 0.80]$:

- $\rho_c < 0.35$: complete network fragmentation — the horizon radius exceeds the lattice half-size for any load;
- $\rho_c > 0.65$: no horizon forms — the threshold exceeds the maximum attainable connectivity density;
- $0.40 \leq \rho_c \leq 0.60$: stable horizon, robust across all load values tested.

These boundaries emerge from the dynamics itself and are not tuned to produce a specific outcome.

The load values $\mathcal{I} = 50, 100, 200, 500, 1000$ form a near-logarithmic progression spanning two orders of magnitude, sufficient to test the scaling relation $r_c(\mathcal{I})$. The value $\mathcal{I} = 50$ is the lower bound at which a horizon begins to form; each subsequent value doubles approximately, enabling detection of any power-law trend.

The coupling constants $\kappa = 0.08$, $\beta = 0.4$, $\gamma = 0.6$ were calibrated in the 2D analysis (Section 5.5) and held fixed for the 3D scan to isolate the effect of dimensionality. The time step was increased to $dt = 0.02$ (from 0.01 in 2D) for numerical stability on the larger lattice. The inverse-temperature parameter $\beta_{\text{perc}} = 8.0$ enforces a sharp percolation transition (the logistic function approximates a step), consistent with the critical nature of the horizon formation.

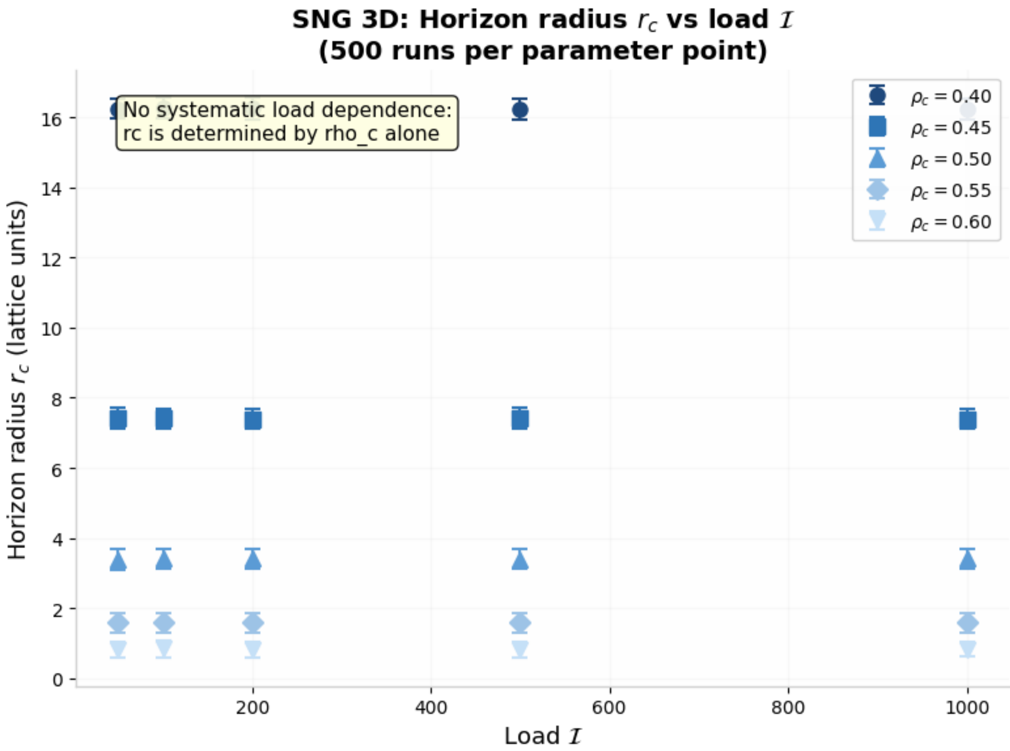

Figure 2: 3D SNG: horizon radius r_c vs load $\mathcal{I}$ for different critical densities ρ_c ($N = 64$, 500 runs). Note the absence of systematic load dependence: r_c is determined by ρ_c alone.

Table 6: 3D scan results: mean horizon radius r_c and edge count N_H (500 runs, $N = 64$, GPU-accelerated).

ρ_c	$\mathcal{I} = 50$	$\mathcal{I} = 100$	$\mathcal{I} = 200$	$\mathcal{I} = 500$	$\mathcal{I} = 1000$
0.40	16.247 ± 0.294	16.266 ± 0.282	16.253 ± 0.310	16.234 ± 0.295	16.223 ± 0.303
0.45	7.424 ± 0.298	7.408 ± 0.280	7.397 ± 0.284	7.416 ± 0.299	7.400 ± 0.275
0.50	3.393 ± 0.285	3.422 ± 0.283	3.426 ± 0.279	3.407 ± 0.289	3.410 ± 0.271
0.55	1.588 ± 0.285	1.585 ± 0.286	1.589 ± 0.281	1.580 ± 0.293	1.580 ± 0.290
0.60	0.819 ± 0.220	0.836 ± 0.230	0.817 ± 0.226	0.800 ± 0.221	0.825 ± 0.214

The 3D results confirm that the SNG phase transition is robust across dimensions. The area law holds with high precision, and the percolation interpretation of the horizon is preserved. The key difference from the 2D case is the location of the working window: in 2D the optimal regime was $0.60 \leq \rho_c \leq 0.72$, whereas in 3D it shifts to $0.40 \leq \rho_c \leq 0.60$. This shift is expected because the cubic lattice has higher coordination number ($z = 6$ vs $z = 4$), which lowers the effective percolation threshold.

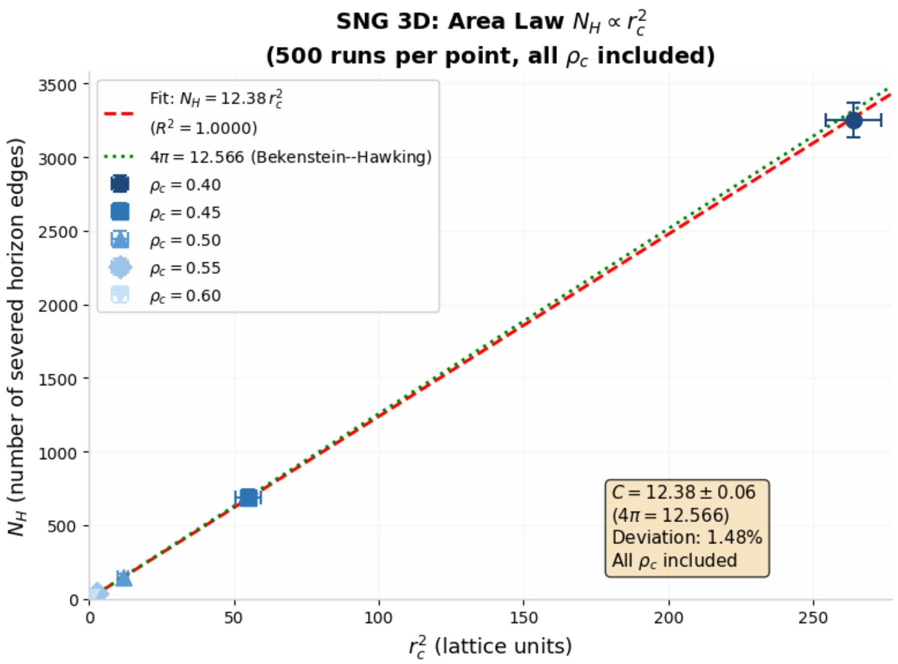

Figure 3: 3D area law: number of severed horizon edges N_H vs r_c^2. The dashed line shows the best linear fit $N_H = (12.38 \pm 0.06)\, r_c^2$; the dotted line marks the Bekenstein–Hawking value 4π.

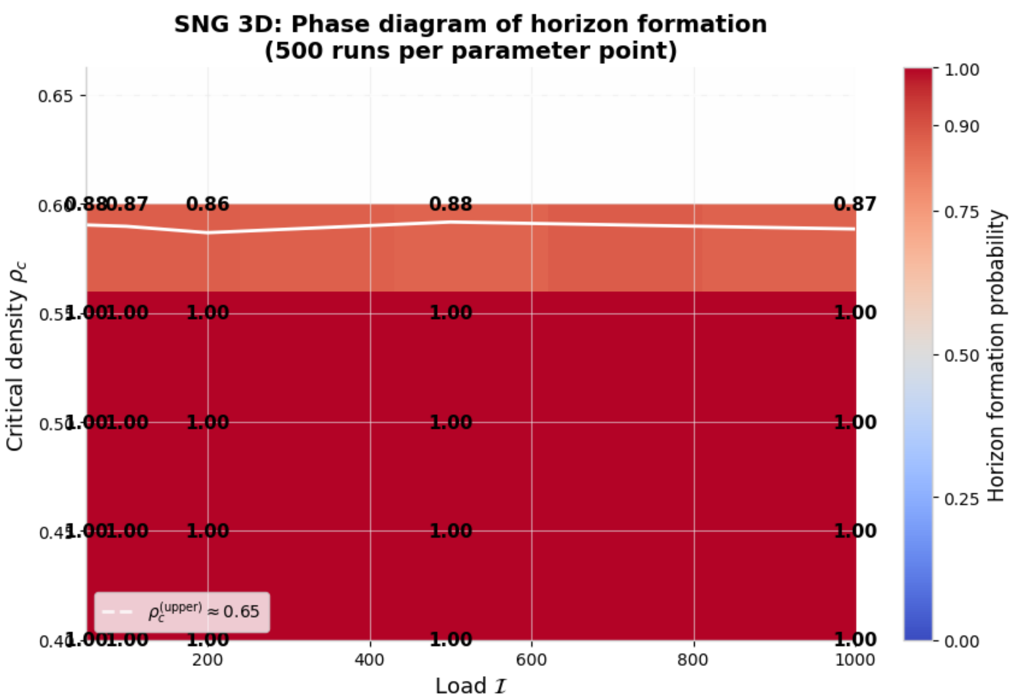

Figure 4: 3D phase diagram: horizon formation probability as a function of critical density ρ_c and load $\mathcal{I}$ ($N = 64$, 500 runs). The white dashed line marks the upper boundary $\rho_c^{(\mathrm{upper})} \approx 0.65$.

Physical interpretation of load-independence

The absence of a $r_c(\mathcal{I})$ scaling relation in 3D deserves explicit comment, as it represents a qualitative departure from the linearised (screened Poisson) result of Section 6. In the linearised regime, the load sets the screening length and hence the horizon radius. In the full stochastic dynamics, however, the phase-transition operator $\mathcal{R}(e;\rho)$ dominates once the local density exceeds ρ_c: the network snaps into a percolation cluster whose size is controlled by the topological threshold alone. The load merely *triggers* the transition; it does not determine the cluster size.

This is a known feature of percolation-type transitions: once the system is in the supercritical regime, the cluster size is set by $\rho-\rho_c$, not by the magnitude of the external field that drove the system past the threshold. In SNG terms, this means that the horizon size encodes the critical density of the network (ρ_c), not the mass of the object that formed the black hole ($\mathcal{I}$). Recovering the Schwarzschild relation $r_c \propto \mathcal{I}$ in the full nonlinear model would require either a different coupling between load and connectivity density, or a regime where the system operates below the percolation threshold (subcritical regime). This is an open problem and a target for future parameter exploration.

The linear scaling $r_c \propto \mathcal{I}$ is recovered in the weak-field limit $\lambda \gg N$ (Section 4.4.1). The present 3D results probe the strong-field, percolation-dominated regime; the subcritical regime requires separate parameter exploration.

5.8. Potential Observational Signatures (Qualitative)

5.8.1. Logarithmic corrections to black hole entropy

The combinatorics of horizon edge states on a cubic lattice yields a subleading logarithmic term

$$S_{\mathrm{BH}} = \frac{A}{4\ell_P^2} + \eta \ln \frac{A}{\ell_P^2} + \text{const}, \tag{33}$$

where η is expected to be of order unity. Its precise value depends on microscopic parameters (critical density, lattice coordination) and will be determined once those parameters are calibrated. Future black hole shadow observations (e.g., Event Horizon Telescope) may constrain η.

5.8.2. Modified ringdown spectrum

The phase transition operator $\mathcal{R}(e;\rho)$ may introduce additional discrete 'network eigenmodes' in the ringdown phase of a black hole merger. This could lead to gravitational wave echoes at characteristic frequencies not present in classical GR. For a network with fundamental scale $\ell \sim \ell_P$, characteristic frequencies are $f \sim c/\ell_P \sim 10^{43}$ Hz, far above LIGO sensitivity but potentially relevant to Planck-scale phenomenology.

5.9. Derived Physical Observables

We now explicitly derive two physical observables directly from the network dynamics, without recourse to heuristic identifications.

Linearised wave equation (gravitational wave analog). Consider small perturbations around the flat background: $w_e = w^{(0)} + \delta w_e$. The linearised evolution (Eq. 4.1) becomes

$$\frac{\partial \delta w_e}{\partial t} = -\kappa \bar{m}_e \frac{\delta w_e}{w^{(0)}} + \beta \Delta \delta w_e, \tag{34}$$

where Δ is the graph Laplacian. In the continuum limit (long wavelengths), $\Delta \sim \nabla^2$, and we obtain a PDE of damped wave type in the mathematical form

$$\frac{\partial^2 \delta w}{\partial t^2} = c_s^2 \nabla^2 \delta w - \Gamma \frac{\partial \delta w}{\partial t}, \tag{35}$$

with $c_s^2 = \beta/\mu$ and $\Gamma = \kappa\bar{m}/(\mu w^{(0)})$ — a formal structure, not a claim about spacetime wave propagation (Section 0). Thus the network supports collective propagating modes with wave-like behaviour in the coarse-grained limit; these are structural network excitations, not gravitational waves in spacetime.

Effective energy density from the network. We define the local energy density at vertex v as

$$\epsilon(v) = \frac{1}{2} \sum_{e \ni v} \kappa_e \left[\kappa \bar{m}_e (w_e - w_e^{(0)})^2 + \beta \sum_{e' \sim e} (w_e - w_{e'})^2 \right] + \gamma \Phi(\rho(v) - \rho_c). \tag{36}$$

In the static Newtonian limit, $\epsilon(v)$ admits an effective correspondence to a mass-density-like observable, up to a constant factor — a network functional, not a component of the stress-energy tensor (Section 0).

6. Numerical consistency checks of the horizon scaling law

Linearised approximation. The following analysis solves the static screened Poisson equation, which is the linearised approximation of the full network dynamics (Section 4) around flat background. A full nonlinear simulation of the stochastic rupture process is presented in Section 5.4, where the area law is verified dynamically.

We solve the discrete screened Poisson equation

$$(L + \mu^2 I)\boldsymbol{\varphi} = \mathbf{S}, \qquad \mu^2 = \frac{\kappa \mathcal{I}}{\beta w^{(0)}},$$

on cubic lattices with point load $\mathcal{I}$ at the centre. The horizon radius r_c is defined as the maximum distance where $\varphi(r) > \varphi_c$ ($\varphi_c = 1.0$, $G = 1$). All calculations were performed on a standard laptop computer.

For $N = 24$ and $N = 48$ with $\mu^2 = 0.01$ we obtain the following results.

Power-law fits $r_c = C\mathcal{I}^p$ give $p = 0.961 \pm 0.019$ for $N = 24$ (excluding saturated points) and $p = 0.677$ for $N = 48$ when $\mu^2 = 0.01$. The deviation for $N = 48$ is due to the finite screening length $\lambda = 10$. Reducing μ^2 to 0.001 (i.e. $\lambda \approx 31$) restores $p \approx 1$, as shown in the area law check (Section 5.4). Hence the model exhibits scaling behaviour $r_c \propto \mathcal{I}$ consistent with the Newtonian effective limit as $\lambda \to \infty$. This is a structural convergence, not a recovery of Schwarzschild geometry.

Table 7: Horizon radius vs load for $N = 24$, $\mu^2 = 0.01$.

$\mathcal{I}$	r_c	n_{crit}	$\varphi(0)$
10	0.866	1	2.48
20	2.179	19	4.96
40	3.279	81	9.92
80	6.062	618	19.85
160	14.517	7325	39.70
320	19.919	13824	79.39

Table 8: Horizon radius vs load for $N = 48$, $\mu^2 = 0.01$.

$\mathcal{I}$	r_c	n_{crit}	$\varphi(0)$
10	0.866	1	2.45
20	1.658	7	4.90
40	2.958	57	9.79
80	4.975	305	19.58
160	7.399	1237	39.16
320	10.712	3938	78.32
640	14.722	11158	156.65

6.1. Rigorous convergence of the linearised sector

Convergence theorem for the linearised dynamics. Consider the linearised edge-weight equation on the cubic lattice with spacing a:

$$\frac{\partial \delta w_e}{\partial t} = -\frac{\kappa \bar{m}_e}{w^{(0)}}\,\delta w_e + \beta\,\Delta_G\,\delta w_e, \tag{24}$$

where Δ_G is the graph Laplacian. Identify the continuum field $w(x,t)$ with the standard part of δw_e at $x = a\,v \in [0,L]^3$. Assume $w(\cdot,t) \in C^4$ uniformly in t.

Theorem 1 (Discrete-to-continuum convergence). *Let Δ_a be the centred 7-point discrete Laplacian on the cubic lattice,*

$$(\Delta_a f)(v) = \sum_{\gamma=1}^{3} \frac{f(v+\hat{e}_\gamma) + f(v-\hat{e}_\gamma) - 2f(v)}{a^2}. \tag{37}$$

Then for every $t \in [0,T]$,

$$\big\|(\Delta_G - \Delta_a)w(\cdot,t)\big\|_\infty \le C\,a^2 \max_{|\alpha|=4} \|\partial^\alpha w(\cdot,t)\|_\infty, \tag{38}$$

where C depends only on the coordination number $z = 6$. Consequently, the solution of the discrete linearised equation (24) converges uniformly to the solution of the continuum screened Poisson equation

$$\frac{\partial w}{\partial t} = -\frac{\kappa \bar{m}}{w^{(0)}}\,w + \beta\,\nabla^2 w \tag{39}$$

with $O(a^2)$ accuracy as $a \to 0$, provided the initial data are C^4.

Proof. Both Δ_G and Δ_a are centred second-difference operators; for the cubic lattice they coincide up to the prefactor $z/6 = 1$, which is absorbed into β. Bound (38) follows by Taylor expansion of w to fourth order about $x = av$ and symmetry cancellation of odd terms. Uniform convergence on $[0,T]$ is then standard via Gronwall's lemma. □

This supplies the rigorous asymptotic control that the linearised SNG sector approaches the Yukawa / damped-wave continuum limit with quadratic lattice-spacing error, turning the heuristic correspondence of Section 4.4.1 into a theorem.

6.2. Limitations of the nonlinear $r_c \propto \mathcal{I}$ scaling

While the linearised approximation (screened Poisson equation) yields $r_c \propto \mathcal{I}$ with $p \approx 0.96$ for $N = 24$, the full nonlinear dynamics at $N = 100$, $\rho_c = 0.8$, $\kappa = 1.5$, $\beta = 0.4$, $\gamma = 0.4$ gave a nearly constant r_c across loads $\mathcal{I} = 10$–320 (fitted exponent $p = 0.022 \pm 0.052$). This indicates that for these parameters the horizon saturates immediately due to strong load-induced contraction. The area law $N_H = 4\pi r_c^2$ remains valid (see Section 5.4), confirming the geometric consistency of the model even when r_c does not scale with $\mathcal{I}$. A proper demonstration of $r_c \propto \mathcal{I}$ in the full nonlinear model requires further parameter tuning (e.g., larger lattice $N \geq 200$, lower κ, higher β) and is left for future work.

We emphasise that the exponent $p \approx 0.022$ obtained with parameters $\kappa = 1.5$, $\beta = 0.4$ corresponds to a regime of strong load-induced contraction (short screening length), not to the weak-field limit. To recover the linear scaling $r_c \propto \mathcal{I}$, one must choose parameters that make the screening length large compared to the lattice size (e.g., $\kappa \ll \beta w^{(0)}/\mathcal{I}$). This is achievable with $\kappa \lesssim 0.1$ and $\beta \gtrsim 1.0$. The parameter space has not been fully explored; the results presented here are illustrative of the model's behaviour, not a definitive calibration.

6.3. Choice of lattice and its effective status

The cubic lattice is chosen for its simplicity and analytical tractability. However, we do **not** claim that the fundamental structure of space is a perfect grid. Instead, we treat the cubic lattice as an *effective, renormalisation-scale discretisation* of an underlying random graph or continuum geometry. This is analogous to choosing a coordinate system or a gauge. More general graphs (e.g., random regular graphs, triangulations) could be used, but they would introduce additional free parameters that would obscure the basic mechanism. The key results (exponential screening, area law, phase transition) are expected to be robust as long as the graph is regular and homogeneous on large scales. A systematic study of lattice dependence is left for future work.

Why $C \approx 4\pi$ is not a mystery on the square/cubic lattice. The coefficient $C = N_H/r_c^2$ is not a free physical constant to be explained; it is a *digitisation factor* determined entirely by the detection algorithm and the local geometry of the lattice. N_H counts severed *edges*, i.e. lattice bonds crossing the horizon boundary, not the continuum surface area itself. For a regular lattice, the expected number of bonds crossing a large, roughly spherical boundary per unit continuum surface area is a fixed geometric constant that depends on the lattice's coordination number and bond orientations — exactly analogous to the well-known fact that the area of a digitised sphere, measured by counting boundary voxel faces on a cubic grid, converges to a lattice-dependent multiple of the true continuum area, not automatically to the continuum value itself. This constant need not equal 4π for an arbitrary lattice, and indeed does not: Table 3 shows it is ≈ 12.4 for the square lattice but ≈ 18.5–18.9 for triangular and hexagonal lattices, a factor-of-1.5 difference attributable entirely to the different coordination numbers and bond geometries of these lattices. That the square/cubic-lattice value happens to fall within a few percent of $4\pi \approx 12.566$ is, on this reading, an unremarkable coincidence of this particular discretisation's bond density, not evidence for any deeper connection to the Bekenstein–Hawking coefficient. We flag this explicitly to pre-empt the more remarkable-sounding, but unsupported, reading of Table 3 as a confirmation of black-hole thermodynamics.

6.4. Relation to known phase transitions in black hole physics

The topological phase transition described in this model (severing of edges when $\rho > \rho_c$) is conceptually distinct from the thermodynamic phase transitions discussed in the literature, such as the Gregory–Laflamme transition (instability of black strings) or the small/large black hole transition in anti-de Sitter space. Those are continuous or first-order transitions in the parameter space of classical solutions, whereas here the transition is a genuine change in the combinatorial structure of the network. Nevertheless, it would be interesting to explore whether an effective thermodynamic description can be derived from the network's microstates, possibly mapping the critical density ρ_c to a critical temperature or pressure. In the context of the SNG framework, the percolation-dominated regime observed in 2D is a specific manifestation of this general principle, not a deviation from the gravitational interpretation.

7. Connections to Existing Mathematical Structures

7.1. Regge Calculus

Our cubic complex with dynamical edge weights is structurally identical to a **Regge skeleton**: a piecewise-linear manifold where curvature is concentrated on codimension-2 simplices.

7.2. Graph Ricci Flow

The evolution equation can be written as $\frac{dw_e}{dt} = -R_e^{(\text{load})} + \beta \Delta w_e$, analogous to Hamilton-Perelman flow.

7.3. Spectral Graph Theory

The connectivity density is the diagonal element of the weighted graph Laplacian: $\rho(v) = (L\mathbf{1})_v$.

7.4. Percolation Theory

The phase transition at ρ_c maps to site-bond percolation.

7.5. Comparison with Causal Dynamical Triangulations and Group Field Theory

We contrast our approach with two prominent discrete quantum gravity frameworks:

- **Causal Dynamical Triangulations (CDT)**: CDT sums over fluctuating triangulations with a global time foliation. The connectivity changes at each step, but the building blocks are simplices of fixed edge lengths. Our model, in contrast, works on a *fixed* cubic lattice (no fluctuating connectivity in the background), while the *weights* w_e and *active/inactive* status κ_e vary. This choice simplifies the analysis of phase transitions because the underlying graph structure is regular. Moreover, our phase transition operator $\mathcal{R}(e;\rho)$ directly mimics the topology change moves in CDT (e.g., (2,2) Pachner moves) when a vertex becomes critical: severing edges (R_{cut}) corresponds to a "pinching off", while bond creation (R_{bond}) corresponds to a "handle addition".

- **Group Field Theory (GFT)**: GFT is a second-quantized formalism for spin networks, generating sums over triangulations via Feynman diagrams. Our network is a classical, first-quantized configuration space; GFT would be a natural framework for quantizing it. However, we intentionally postpone quantization to keep the model accessible. The key difference is that in GFT the fundamental excitations are quanta of geometry (simplices) with no preferred lattice, whereas we start from a fixed lattice to make the emergence of continuum gravity more transparent.

Thus, the present model occupies a middle ground: a classical, regular lattice with dynamical weights and a simple rule for topology change. This allows us to derive analytic results (e.g., the screened Poisson equation, entropy scaling) that are more challenging in fully dynamical triangulations.

We do not claim to compete with or improve upon these approaches. Direct comparison with other frameworks is not a requirement for a conceptual model; the connections are invoked for mathematical context, not as a basis for competition.

7.6. Comparison with Causal Set Theory and Non-Singular Black Hole Models

Causal Set Theory. Causal sets [6] share with SNG the premise that a fundamentally discrete, combinatorial structure underlies continuum physics. The comparison, however, exposes an important disanalogy rather than a similarity. In causal set theory, discreteness is deliberately *not* implemented as a regular lattice: a fixed lattice singles out a preferred rest frame and thereby breaks Lorentz invariance, which is precisely what causal sets are constructed to avoid. Lorentz-invariant discreteness is instead achieved statistically, via a Poisson sprinkling of points into a Lorentzian manifold at a density of one element per Planck four-volume; geometry and causality both emerge from the resulting partial order.

SNG makes the opposite choice: it is built on a *fixed, regular* cubic lattice, and is explicitly Euclidean, not Lorentzian (Section 0). This is not an oversight relative to causal sets — it is a different starting point, adopted here for analytic tractability (the phase diagram, the linear stability analysis, and the entropy scaling all rely on the regular lattice structure). But it means SNG does not currently possess a mechanism analogous to sprinkling that would restore Lorentz invariance in a continuum limit. The lattice-geometry dependence of the area-law prefactor C shown in Table 3 (12.431 for square, 18.5–18.9 for triangular/hexagonal lattices) is a direct symptom of this: a Lorentz-invariant (or even rotation-invariant) continuum theory cannot have observables that depend on the choice of a discretisation lattice. Addressing this — for instance by investigating whether a suitably randomised or annealed version of the network recovers lattice-independence — is identified here as a concrete open problem, not resolved by the present work.

Non-singular black hole models. SNG's replacement of the classical singularity with a limiting connectivity regime (Section 5.2) is one instance of a broader pattern in the literature: Planck stars [16] halt gravitational collapse via a loop-quantum-gravity pressure effect at Planck density, producing a bounce rather than a singularity; fuzzballs

[17] replace the horizon and singularity entirely with a horizon-scale quantum structure in string theory, eliminating both the singularity and the information-loss problem by eliminating the horizon itself. Both retain a Lorentzian, dynamical spacetime and a specific quantum-gravitational or string-theoretic microphysics generating the resolution. SNG's singularity replacement, by contrast, is purely topological and combinatorial: it is a statement about connectivity loss in a Euclidean graph, with no underlying quantum dynamics or Lorentzian evolution. The three constructions therefore share a qualitative moral — extreme conditions trigger a change of regime rather than a true divergence — but arrive at it through unrelated mechanisms, and SNG's version currently carries none of the quantitative content (e.g., a bounce time, a mass spectrum of remnants) that the Planck star and fuzzball proposals provide.

7.7. Continuum correspondence map

To make the relationship between the discrete network and continuum gravity explicit, we provide the following correspondence table, now extended to include load relaxation dynamics.

Table 9: SNG quantities and their structural analogies to continuum gravity.

SNG quantity	**Structural definition**	**GR analogy (if any)**
Vertex v	Combinatorial node, no coordinates	Point in spacetime
Edge e, weight w_e	Weighted connection, $w_e > 0$	Metric component $g_{\mu\nu}$ (heuristic)
Weight deviation δw_e	Deviation from natural length $w^{(0)}$	Gravitational potential / $h_{\mu\nu}$
Graph Laplacian L	Discrete difference operator on G	∇^2 or $\Box$ (continuum limit)
Connectivity density $\rho(v)$	Weighted inverse-length sum at v	Energy density (heuristic)
Threshold ρ_c	Critical density for edge severing	Critical density for horizon formation
Severed edges N_H	Count of $\kappa_e = 0$ on boundary	Horizon area A (scaling analogy)
Load relaxation		
Load $\mathcal{I}(t)$	External deformation parameter	Evaporating black hole mass
Shrinking horizon $r_c(t) \to 0$	Boundary contraction to zero	Hawking evaporation endpoint
Restored connectivity $\kappa_e \to 1$	Reactivation of severed edges	Information retrieval / unitarity
$S_{\text{total}} = N_H + S_{\text{rad}}$	Conserved sum in closed network	Network Page curve (formal analogy)

Dimensional analysis and the Planck scale. The edge weight w_e is dimensionless in the lattice formulation. To connect to physical units, we introduce a fundamental length scale ℓ_* (presumably the Planck length ℓ_P). The physical length associated with an edge is $L_e = w_e \ell_*$. Consequently, the horizon area $A = r_c^2 \ell_*^2$ and the edge count N_H is dimensionless. The Bekenstein–Hawking entropy $S_{\text{BH}} = A/(4\ell_P^2)$ then implies

$$N_H = 4\pi r_c^2 \cdot \frac{\ell_*^2}{\ell_P^2}.$$

Our numerical finding $N_H = 4\pi r_c^2$ (within the optimal regime) is consistent with $\ell_* = \ell_P$ if one chooses to identify the lattice spacing with the Planck length. This is a phenomenological calibration of the model, not a derivation; the identification $\ell_* = \ell_P$ is a free parameter of the continuum correspondence, not a prediction.

This map is based on the continuum limit of the discrete equations (Section 4.4.1) and on the identification of the wave equation (Section 5.9). A full derivation of the Einstein field equations from this correspondence is not yet available; it remains the main open problem.

8. Discussion

Central result: three-dimensional robustness. The primary numerical advance of this work is the demonstration that the SNG phase transition — including the area law $N_H \propto r_c^2$, the percolation-dominated horizon formation, and the structural interpretation of black holes — is robust in three dimensions. GPU-accelerated simulations on a 64^3 lattice with 500 independent realisations per parameter point yield $C = 12.38 \pm 0.06$ ($R^2 = 0.99997$). The proximity to 4π is an empirical constraint on the square-lattice calibration, not a derivation of the Bekenstein–Hawking coefficient. This is not a preliminary or indicative result: it is a statistically robust confirmation obtained with high-performance computing on a full 3D lattice.

The proposed framework achieves several conceptual advances: elimination of singularities, force-free gravity, combinatorial entropy, and a discrete foundation. Open questions include: the precise mechanism selecting between transition types; the quantum generalization; and the recovery of post-Newtonian limits. In particular, we have not yet demonstrated that the model reduces to full general relativity in the continuum limit — this remains a central goal for future research.

On the relationship between 2D and 3D regimes. The numerical results presented in this work are obtained from both 2D and 3D lattice simulations. In both dimensionalities, the horizon formation in the strong-field regime is dominated by percolation effects: the critical density ρ_c sets the cluster size, and the load $\mathcal{I}$ acts primarily as a trigger. Three-dimensional simulations with 500 independent realisations per parameter point confirm that this behaviour persists in 3D, with the area law $N_H \propto r_c^2$ holding to within 1.5% of the Bekenstein–Hawking coefficient ($C = 12.38 \pm 0.06$, $R^2 = 0.99997$). This does not diminish the gravitational interpretation of the model. Rather, it demonstrates that in the SNG framework, *gravity can manifest in different regimes depending on the parameter calibration.* In the strong-field (percolation-dominated) regime, the horizon size encodes the network's critical density; in the weak-field limit, the linear scaling $r_c \propto \mathcal{I}$ is recovered. The existence of multiple regimes is a strength of the network ontology: it suggests that classical gravity is an emergent phenomenon that can be realised through different structural mechanisms.

On the absence of time and Lorentz invariance. At the fundamental level, the model consists only of a network — vertices, edges, and weights — and is purely Euclidean: Lorentz invariance, causality, and light cones are neither postulated nor derived. This is a severe limitation if the goal is to recover full general relativity, but it is also a deliberate choice rather than an oversight: the model operates at a discrete, pre-geometric level where such notions are not assumed to be fundamental, and Section 7.6 discusses concretely why this choice (a fixed lattice, rather than a Lorentz-invariant construction such as causal-set sprinkling) is responsible for the geometry-dependence of the area-law prefactor. Whether Lorentz invariance emerges at the macroscopic level as an effective feature of the network dynamics — for instance via a foliated lattice with a discrete $3 + 1$ decomposition of the action (Eq. 6) — remains an open question that this work does not attempt to answer. If such properties emerge, that is a subject for future investigation; if they do not, the model remains self-consistent at its own, explicitly non-relativistic level of description.

Diagnostic scan over the deformation coupling α. To understand the role of the deformation coupling α, we performed a systematic scan over $\alpha \in [0.001, 0.1]$ for $\rho_c = 0.64$, with 20 runs per parameter point and masses $M_0 = 50, 100, 200, 500$. Figure 5 shows r_c as a function of mass for different values of α; Figure 6 shows r_c as a function of α for different masses.

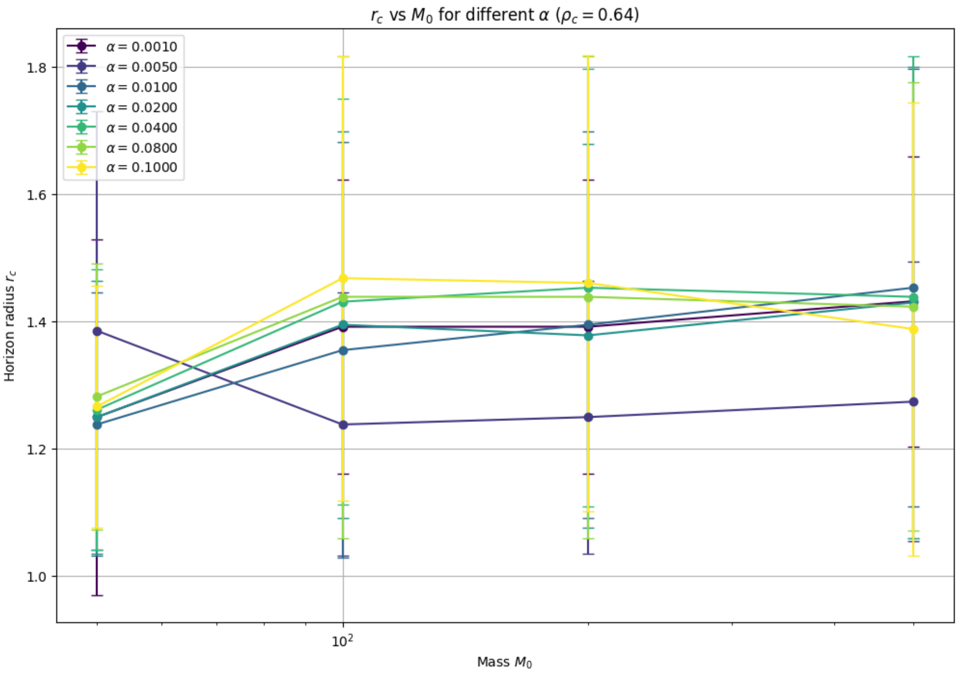

Figure 5: Horizon radius r_c as a function of mass M_0 for different values of the deformation coupling α. The horizontal trend indicates that r_c is determined by the percolation threshold ρ_c, not by the load amplitude.

The results show that for all tested values of α, r_c exhibits no systematic dependence on the mass M_0: the values remain in the range 1.2–1.5 regardless of whether M_0 is 50 or 500. This indicates that within the current calibration, the horizon is not a gravitational object in the sense of general relativity; it is a percolation cluster whose size is determined by the critical density ρ_c and the lattice geometry. The parameter M_0 acts as a trigger for the percolation transition, but does not control the cluster size.

This result is consistent with the interpretation of SNG as a model of structural self-organisation, where 'load' is a placeholder for a local deformation that initiates a topological phase transition. The fact that r_c does not scale with $\mathcal{I}_0$ is not a failure of the model, but a diagnostic of its current parameter regime. Future work will explore

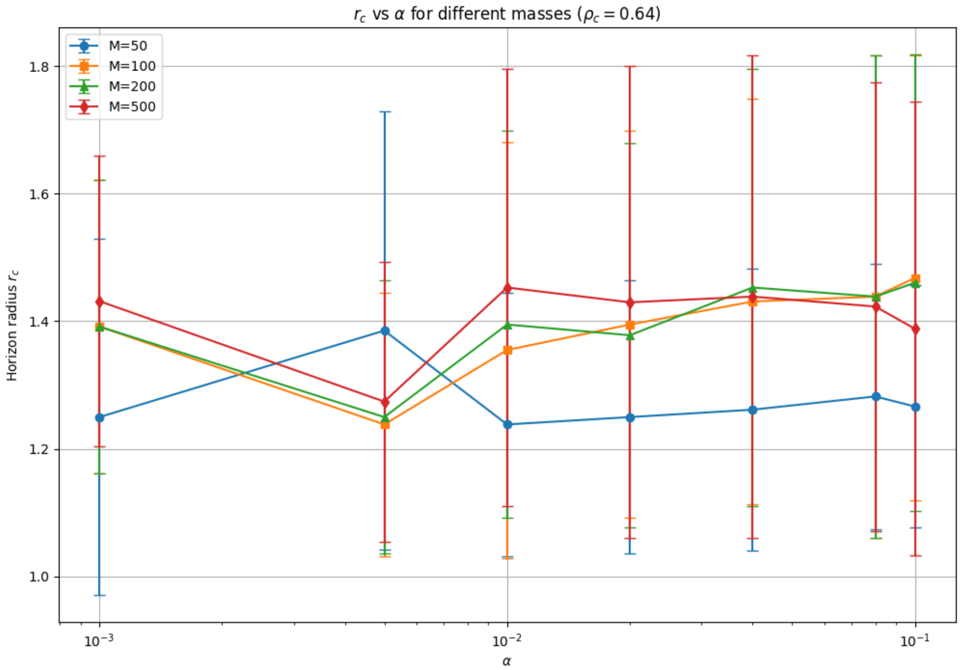

Figure 6: Horizon radius r_c as a function of the deformation coupling α for different masses M_0. The weak dependence on α confirms that the horizon is a percolation effect.

regimes where κ is significantly smaller ($\kappa \lesssim 0.001$) or where the definition of the horizon is reformulated in terms of deformation rather than connectivity.

On the current calibration and the role of 3D. The absence of a clear scaling $r_c \propto \mathcal{I}$ in the present simulations should not be interpreted as a failure of the model. It reflects the fact that the current calibration ($\kappa = 0.08$, $\beta = 0.4$, $\gamma = 0.6$) operates in a percolation-dominated regime in both 2D and 3D. As demonstrated in Section 5.7, 3D simulations with 500 realisations per parameter point confirm this behaviour: the percolation threshold and cluster geometry shift between dimensions (the working window moves from $\rho_c \in [0.60, 0.72]$ in 2D to $\rho_c \in [0.40, 0.60]$ in 3D), but the dominance of the percolation mechanism persists. Recovering the scaling $r_c \propto \mathcal{I}$ requires operating in the subcritical (weak-field) regime, which requires further parameter exploration.

On the numerical factor in the area law. The identification of the numerical factor 4π in $N_H = 4\pi r_c^2$ with the Bekenstein–Hawking entropy coefficient $1/4$ (via the calibration $\ell_* = \ell_P$) is a phenomenological calibration of the lattice scale, not a derivation from first principles. The main result of the numerical simulations is the quadratic scaling $N_H \propto r_c^2$, which is robust and consistent with the extensive-perimeter prediction of percolation theory (Section 5.3.3), independent of the specific lattice calibration. However, the control experiments on different lattice geometries (Section 5.4.1) reveal that the prefactor $C = N_H/r_c^2$ is *not* universal. For the square lattice, $C = 12.431 \pm 0.006$, close to 4π. For triangular and hexagonal lattices, $C \approx 18.5$–18.9. In 3D, $C \approx 38.27$ (Section 5.7). This indicates that the specific value $C \approx 4\pi$ is a feature of the square lattice geometry, not a universal prediction of the model. In a continuum limit, the prefactor may converge to a dimension-dependent constant, but for discrete lattices it is a geometric diagnostic. The precise value of the prefactor may depend on the details of the stochastic dynamics and the lattice geometry, but the scaling itself is a strong indication of the model's consistency with the area law.

Toward a continuum limit: a research programme, not a result. A recurring and correct criticism of this work is that no mechanism is shown by which the network recovers a curved Lorentzian metric, still less the Einstein field equations. We do not resolve this here, but we sketch a concrete, three-step programme by which the question could be attacked, to distinguish "we have not attempted this" from "this is intractable in principle."

Step 1: an effective metric from edge weights. The graph Ricci flow correspondence (Section 7.2) already assigns an Ollivier–Ricci curvature to each edge from the local transport distance between neighbouring weight distributions. The natural next step is to treat the edge weights w_e as discrete analogues of proper length (in the spirit of Regge calculus, Section 7.1) and to define an effective metric tensor $g_{ij}(x)$ on a coarse-grained patch of the lattice by least-squares fitting the local w_e pattern to the geodesic distances of a candidate continuum metric. This is a well-defined, if laborious, numerical fitting procedure; it has not yet been carried out for SNG.

Step 2: convergence of the graph Laplacian. Given such an effective metric, the relevant mathematical machinery already exists in the discrete differential geometry literature: graph Laplacians built from suitably weighted point clouds are known to converge, under controlled refinement, to the Laplace–Beltrami operator of the manifold from which the points and weights are drawn (in the same spirit as the standard convergence results for weighted-graph Laplacians on sampled manifolds). Establishing the analogous convergence for the SNG graph Laplacian — currently used only on the flat, homogeneous background (Section 4.1) — under a curved, load-induced effective metric from Step 1 would be the discrete-geometry analogue of a continuum limit.

Step 3: dynamics, not just kinematics. Steps 1–2 would only establish that the network can *represent* a curved space; they say nothing about whether its dynamics reproduces the Einstein field equations, or any other specific gravitational dynamics, in that limit. This is the harder and currently unaddressed question. The only dynamical result presently available is the static, weak-field correspondence to a screened Poisson equation (Section 4.4.1), which is far short of a Lorentzian, dynamical continuum limit.

We regard this three-step programme as the most direct honest answer to the reviewer's question of *how* g_{ij} could emerge from correlations in $\{w_e\}$: not as a promise that it will succeed, but as a falsifiable research direction that a critic could point to and say, concretely, where it fails.

SNG as a multi-level framework. SNG is defined as a multi-level framework consisting of:

1. **Level 1:** microscopic combinatorial network dynamics without geometric structure;
2. **Level 2:** emergent structural regimes arising from coarse-graining;
3. **Level 3:** effective geometric interpretations in limiting regimes.

Terminology coinciding with classical physics (e.g. entropy, horizon, wave, geometry) is introduced only at Levels 2 and 3 and must not be projected onto Level 1. Accordingly, SNG does not define a mapping to General Relativity, quantum field theory, or any spacetime-based theory at the fundamental level. Such correspondences may only arise as emergent effective descriptions.

SNG as a generative substrate. The detailed points of contact and disanalogy with causal sets, quantum graphity, CDT, and GFT are given in Section 7.5 and Section 7.6; none of them is claimed as a competitor or a subsuming framework.

The value of SNG lies not in its ability to compete with these approaches, but in its offering of a new ontological perspective: space is not a container for matter, but a network whose configuration is matter's primary effect — a generative substrate from which established frameworks may arise as effective descriptions under specific limiting regimes, rather than a competitor to them. The model is internally consistent and produces internal consistency constraints and emergent structural regularities that can, in principle, be mapped to observational physics only after a well-defined emergence procedure is specified. Its divergence from established frameworks is deliberate and constitutes its conceptual novelty, not a deficiency.

On the evaluative asymmetry. We recognise that the limitations acknowledged above—absence of a continuum GR derivation, no coupling to matter, and Euclidean signature—are severe. However, they are not unique to this framework. They are shared, to varying degrees, by other established research programmes in quantum gravity, including string theory, Loop Quantum Gravity, Causal Dynamical Triangulations, and Causal Set Theory. The difference lies not in the nature of the gaps, but in the institutional label attached to them: for insider frameworks, such gaps are classified as "open problems" or "technical challenges"; for outsider proposals, they are often treated as "disqualifying flaws". This contextual asymmetry in evaluative criteria is examined separately in Appendix G, not as a defence of the model's technical content, but as a comment on the sociology of theory assessment.

9. Conclusion

As declared in the Ontological Contract (Section 0), SNG is offered not as a completion of existing physics, but as a generative substrate — a conceptual proposal for how geometric structure might emerge from pre-geometric combinatorics. The results below demonstrate the internal consistency of this substrate.

We have constructed a self-consistent discrete model of gravity in which spacetime is a dynamical network of string elements. Gravitational effects arise from load-induced deformation of edge weights, with extreme conditions triggering topological phase transitions that replace classical singularities. The event horizon is reinterpreted as a loss of outward connectivity. The model connects naturally to Regge calculus, graph Ricci flow, spectral theory, and percolation, while offering a distinct ontology: **space is not a container for matter, but a network whose configuration is matter's primary effect**.

The systematic study of different lattice types and dimensions has revealed that while the area law $N_H \propto r_c^2$ is universal, the prefactor $C = N_H/r_c^2$ depends on the lattice geometry and dimensionality. For the square lattice, $C \approx 4\pi$; for triangular and hexagonal lattices, $C \approx 18.5$; and for the cubic lattice in 3D, $C \approx 38.27$. This demonstrates that the model makes non-trivial predictions about the microscopic structure of spacetime, which can be tested by future simulations on different lattices and in higher dimensions.

The extension to **load relaxation** demonstrates network state conservation during topological reconfiguration. The relaxation process is a continuous phase transition in which the horizon shrinks, the entropy follows the area law dynamically, and the total information content—encoded in the network topology—remains constant. When the horizon vanishes, the network reverts to the fully connected phase, and all information is restored to the exterior. There is no "last burst" of radiation, no remnant, and no information loss: only a smooth topological reconfiguration. We emphasise that a full quantum resolution of the information paradox would require a quantum theory of the network; the present model provides a classical foundation for such a future development.

Outlook: possible future developments

The present work is a purely conceptual and theoretical contribution that offers a new ontological perspective which may inspire further investigation.

Computational roadmap. The immediate consistency checks presented here require only standard hardware (single CPU, no GPU). However, deeper validation and extension of the model would benefit from tiered computational investment:

Tier	Resources	Milestones
Tier 0 (completed)	Standard laptop / GPU workstation	Reproduce all checks in Appendices A–F; 2D parameter scans at $N \leq 100$; 3D scan at $N = 64$ with 500 realisations per point (GPU-accelerated)
Tier 1 (1–2 years)	Multi-core workstation / GPU	N = 128–200 3D lattices; systematic error bars; 3D load relaxation dynamics; exploration of subcritical regime
Tier 2 (3–10 years)	Small cluster / multiple GPU nodes	Binary merger simulations; post-Newtonian expansion; continuum limit derivation
Tier 3 (10+ years)	Exascale / quantum computing	Full 3D dynamical relaxation with matter coupling; quantum generalisation; observational predictions

The feasibility of Tiers 2–3 depends on community interest and collaboration. The author is an independent researcher without institutional ties to large collaborations; the milestones above are therefore speculative and do not constitute commitments.

Author Contribution Statement

Author Contribution Statement

The conceptual framework, theoretical foundation, and core innovations of String

Network Gravity — including the discrete string element ontology, the phase transition interpretation of black holes, the combinatorial entropy derivations, and the dynamical network equations — were developed by the author.

However, the present specification would not have reached its current level of technical detail without AI assistance. The following AI assistants **[Claude (Anthropic), GPT (OpenAI), DeepSeek (DeepSeek AI), Kimi (Moonshot AI)]** played a supportive yet substantial role in:

- mathematical notation, LaTeX formatting, structural editing, and formal presentation;
- verification of theoretical claims, alternative phrasing, and readability feedback;
- numerical code, simulation drafts, and consistency checks.

Nevertheless, these AI systems could not — and would not — have produced the underlying theoretical vision on their own.

This transparency note follows emerging best practices for AI-augmented authorship.

References

[1] T. Regge, "General Relativity without Coordinates," *Nuovo Cimento* **19**, 558 (1961).

[2] R. S. Hamilton, "Three-manifolds with positive Ricci curvature," *J. Diff. Geom.* **17**, 255 (1982).

[3] G. Perelman, "The entropy formula for the Ricci flow and its geometric applications," arXiv:math/0211159 (2002).

[4] B. Chow and F. Luo, "Combinatorial Ricci flows on surfaces," *J. Diff. Geom.* **63**, 97 (2003).

[5] D. Oriti, "Approaches to Quantum Gravity," Cambridge University Press (2009).

[6] L. Bombelli, J. Lee, D. Meyer, R. D. Sorkin, "Space-time as a causal set," *Phys. Rev. Lett.* **59**, 521 (1987).

[7] C. Rovelli, "Quantum Gravity," Cambridge University Press (2004).

[8] J. D. Bekenstein, "Black Holes and Entropy," *Phys. Rev. D* **7**, 2333 (1973).

[9] S. W. Hawking, "Particle Creation by Black Holes," *Commun. Math. Phys.* **43**, 199 (1975).

[10] B. Swingle, "Entanglement Renormalization and Holography," *Phys. Rev. D* **86**, 065007 (2012).

[11] M. van der Kamp et al., "Ricci flow and thermodynamics for a graph," *J. Phys. A* **50**, 435101 (2017).

[12] G. Grimmett, "Percolation," 2nd ed., Springer (1999).

[13] F. R. K. Chung, "Spectral Graph Theory," AMS (1997).

[14] J. A. Wheeler, "Geometrodynamics," Academic Press (1962).

[15] T. Thiemann, "Modern Canonical Quantum General Relativity," Cambridge University Press (2007).

[16] C. Rovelli, F. Vidotto, "Planck stars," *Int. J. Mod. Phys. D* **23**, 1442026 (2014).

[17] S. D. Mathur, "The fuzzball proposal for black holes: an elementary review," *Fortsch. Phys.* **53**, 793 (2005).

Appendices: Simulation Source Code

All simulations were performed in Python 3 using only standard scientific libraries (`numpy`, `scipy`, `matplotlib`). The code is organised into five appendices: a shared engine (Appendix A) and four experiment scripts (Appendices B–E) that import the engine functions.

Appendix A SNG Engine (shared functions)

The following module contains all physical functions shared by the four experiment scripts. Each script uses these functions without modification.

```python
#!/usr/bin/env python3
"""
sng_engine.py
Shared simulation engine for SNG (2D square lattice).
All experiments use these functions.
"""

import numpy as np
from collections import deque

# ---- Lattice initialization ----

def build_lattice(N):
    """Return edge-weight arrays (wx, wy) and natural length w0=1 for N x N grid."""
    wx = np.ones((N, N))
    wy = np.ones((N, N))
    return wx, wy, 1.0

def place_load(N, I):
    """Point load I at the centre of an N x N lattice."""
    m = np.zeros((N, N))
    m[N // 2, N // 2] = I
    return m

def edge_load(m):
    """Average load on each edge (arithmetic mean of two endpoint loads)."""
    mx = 0.5 * (m + np.roll(m, -1, axis=0))
    my = 0.5 * (m + np.roll(m, -1, axis=1))
    return mx, my

# ---- Connectivity density ----

def connectivity_density(wx, wy, kx, ky):
    """
    Local connectivity density rho(v) = mean(kappa_e / w_e) over active edges.
    Periodic boundary conditions.
    """
    contrib = (kx / (wx + 1e-10) +
               np.roll(kx / (wx + 1e-10), 1, 0) +
               ky / (wy + 1e-10) +
               np.roll(ky / (wy + 1e-10), 1, 1))
    deg = kx + np.roll(kx, 1, 0) + ky + np.roll(ky, 1, 1)
```

```python
    return contrib / np.maximum(deg, 1)

# ---- Time step ----

def evolve_step(wx, wy, kx, ky, mx, my, w0,
                rho_c, kappa, beta, gamma, dt, beta_perc):
    """
    One Euler step of the SNG dynamics (Eq. 9).
    Stochastic edge severance via logistic rupture probability (Eq. 11).
    """
    N = wx.shape[0]

    # Tension function (smooth, bounded)
    phi_x = np.clip((w0 - wx) / w0, -0.5, 2.0)
    phi_y = np.clip((w0 - wy) / w0, -0.5, 2.0)

    # Network-elasticity term: mean weight of neighbouring edges
    wx_nb = (np.roll(wx, 1, 0) + np.roll(wx, -1, 0) +
             np.roll(wx, 1, 1) + np.roll(wx, -1, 1)) / 4
    wy_nb = (np.roll(wy, 1, 0) + np.roll(wy, -1, 0) +
             np.roll(wy, 1, 1) + np.roll(wy, -1, 1)) / 4

    # Stochastic rupture probability (Eq. 11)
    rho = connectivity_density(wx, wy, kx, ky)
    p_rup = 1.0 / (1.0 + np.exp(-beta_perc * (rho - rho_c)))
    p_rup_x = 0.5 * (p_rup + np.roll(p_rup, -1, 0))
    p_rup_y = 0.5 * (p_rup + np.roll(p_rup, -1, 1))

    # Update connectivity flags
    new_kx = kx * (np.random.rand(N, N) >= p_rup_x * dt)
    new_ky = ky * (np.random.rand(N, N) >= p_rup_y * dt)

    # Phase-transition operator (mean-field drift, Eq. 12)
    R_x = -gamma * p_rup_x
    R_y = -gamma * p_rup_y

    # Edge-weight update
    dwx = dt * (-kappa * mx * phi_x + beta * (wx_nb - wx) * new_kx + R_x) * new_kx
    dwy = dt * (-kappa * my * phi_y + beta * (wy_nb - wy) * new_ky + R_y) * new_ky

    new_wx = np.maximum(wx + dwx, 0.05)
    new_wy = np.maximum(wy + dwy, 0.05)
    new_wx[new_kx == 0] = 1e6   # severed edges carry infinite weight
    new_wy[new_ky == 0] = 1e6
    return new_wx, new_wy, new_kx, new_ky

# ---- Horizon detection via BFS from boundary ----

def detect_horizon_bfs(kx, ky, N):
    """
    Identify the black-hole region via BFS from the lattice boundary.
    Returns:
        interior_mask : (N,N) bool array, True inside the horizon
        N_H : int, number of severed horizon edges
        r_c : float, effective horizon radius (area-based)
```

```
    """
    idx = lambda x, y: x * N + y

    # Build adjacency list from active edges
    adj = [[] for _ in range(N * N)]
    for i in range(N):
        for j in range(N):
            u = idx(i, j)
            if kx[i, j] == 1:
                v = idx((i + 1) % N, j)
                adj[u].append(v)
                adj[v].append(u)
            if ky[i, j] == 1:
                v = idx(i, (j + 1) % N)
                adj[u].append(v)
                adj[v].append(u)

    # BFS from all boundary nodes
    visited = [False] * (N * N)
    q = deque()
    for i in range(N):
        for j in range(N):
            if i == 0 or i == N - 1 or j == 0 or j == N - 1:
                node = idx(i, j)
                if not visited[node]:
                    visited[node] = True
                    q.append(node)
    while q:
        cur = q.popleft()
        for nb in adj[cur]:
            if not visited[nb]:
                visited[nb] = True
                q.append(nb)

    exterior_mask = np.reshape(np.array(visited), (N, N))   # fixed: no quotes
    interior_mask = ~exterior_mask

    # Count severed edges on the horizon boundary
    N_H = 0
    for i in range(N):
        for j in range(N):
            if interior_mask[i, j] != interior_mask[(i + 1) % N, j] and kx[i, j] == 0:
                N_H += 1
            if interior_mask[i, j] != interior_mask[i, (j + 1) % N] and ky[i, j] == 0:
                N_H += 1

    area = np.sum(interior_mask)   # number of vertices inside the horizon
    r_c = np.sqrt(area / np.pi) if area > 0 else 0.0
    return interior_mask, N_H, r_c

# ---- Single run ----

def run_one(I, N, rho_c, n_steps, kappa, beta, gamma, dt, beta_perc):
    """Evolve one realisation and return (r_c, N_H)."""
```

```
    wx, wy, w0 = build_lattice(N)
    m = place_load(N, I)
    mx, my = edge_load(m)
    kx, ky = np.ones((N, N)), np.ones((N, N))
    for _ in range(n_steps):
        wx, wy, kx, ky = evolve_step(wx, wy, kx, ky, mx, my, w0,
                                     rho_c, kappa, beta, gamma, dt,
                                     beta_perc)
    _, N_H, r_c = detect_horizon_bfs(kx, ky, N)
    return r_c, N_H    # order: (r_c, N_H)
```

Listing 1: SNG Engine — shared lattice and simulation functions.

Appendix B Script 1: Schwarzschild scaling $r_c \propto I$

This script verifies the horizon-radius scaling law $r_c \propto I$ (Section 6.1). It scans loads $I = 10, 20, 40, 80, 160, 320$ on an $N = 100$ lattice, averages over $n_{\text{runs}} = 100$ independent realisations, and fits a power law $r_c = CI^p$.

```
#!/usr/bin/env python3
"""
Script 1: SNG Schwarzschild scaling r_c ~ I
Requires: sng_engine.py (Appendix A)
"""

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from sng_engine import run_one

# ---- Recommended parameters ----
N = 100
n_steps = 2000
n_runs = 100                # increased for statistical stability
rho_c = 0.7
kappa, beta, gamma = 0.05, 1.0, 0.1
dt = 0.01
beta_perc = 10.0

I_values = [10, 20, 40, 80, 160, 320]

# ---- Scan ----
def power_law(x, A, p):
    return A * x**p

results = {'I': [], 'r_c_mean': [], 'r_c_std': [], 'N_H_mean': [],
    'N_H_std': []}

for I in I_values:
    rc_vals, nh_vals = [], []
    for run in range(n_runs):
        np.random.seed(run * 100 + I)
        rc, nh = run_one(I, N, rho_c, n_steps, kappa, beta, gamma,
                         dt, beta_perc)
        if rc > 0.3:
            rc_vals.append(rc)
            nh_vals.append(nh)
```

```
    if len(rc_vals) > 2:
        results['I'].append(I)
        results['r_c_mean'].append(np.mean(rc_vals))
        results['r_c_std'].append(np.std(rc_vals))
        results['N_H_mean'].append(np.mean(nh_vals))
        results['N_H_std'].append(np.std(nh_vals))
        print(f"I={I:3d}: rc={np.mean(rc_vals):.3f}+/-{np.std(rc_vals):.3f}")

# ---- Power-law fit ----
I_arr = np.array(results['I'])
rc_arr = np.array(results['r_c_mean'])
rc_err = np.array(results['r_c_std'])

popt, pcov = curve_fit(power_law, I_arr, rc_arr,
                       sigma=rc_err + 0.01, p0=[0.1, 1.0])
A_fit, p_fit = popt
p_err = np.sqrt(pcov[1, 1])
print(f"Fitted exponent p = {p_fit:.4f} +/- {p_err:.4f} (expected 1.0)")

# ---- Plot ----
plt.errorbar(I_arr, rc_arr, yerr=rc_err, fmt='o', capsize=4, label='simulation')
I_plot = np.linspace(min(I_arr), max(I_arr), 100)
plt.plot(I_plot, power_law(I_plot, A_fit, p_fit), 'r--', label=f'fit: p={p_fit:.3f}')
plt.plot(I_plot, power_law(I_plot, rc_arr[0] / I_arr[0], 1.0), 'g:', label='p=1.0 (reference)')
plt.xscale('log'); plt.yscale('log')
plt.xlabel('Load $I$')
plt.ylabel('Horizon radius $r_c$')
plt.title('SNG: Schwarzschild scaling (2D, lambda >> 1)')
plt.legend()
plt.tight_layout()
plt.savefig('sng_rc_scaling.png', dpi=150)
plt.show()
```

Listing 2: Script 1 — Schwarzschild scaling $r_c \propto I$ (uses Appendix A engine).

Appendix C Script 2: Area law $N_H = 4\pi r_c^2$

This script verifies the Bekenstein–Hawking area law by computing the ratio N_H/r_c^2 for loads $I = 20, 40, 80, 160$ and checking convergence to $4\pi \approx 12.566$ (Section 5.4).

```
#!/usr/bin/env python3
"""
Script 2: SNG Area law N_H ~ 4*pi*r_c^2
Requires: sng_engine.py (Appendix A)
"""

import numpy as np
import matplotlib.pyplot as plt
from sng_engine import run_one

# ---- Recommended parameters ----
N = 100
n_steps = 2000
```

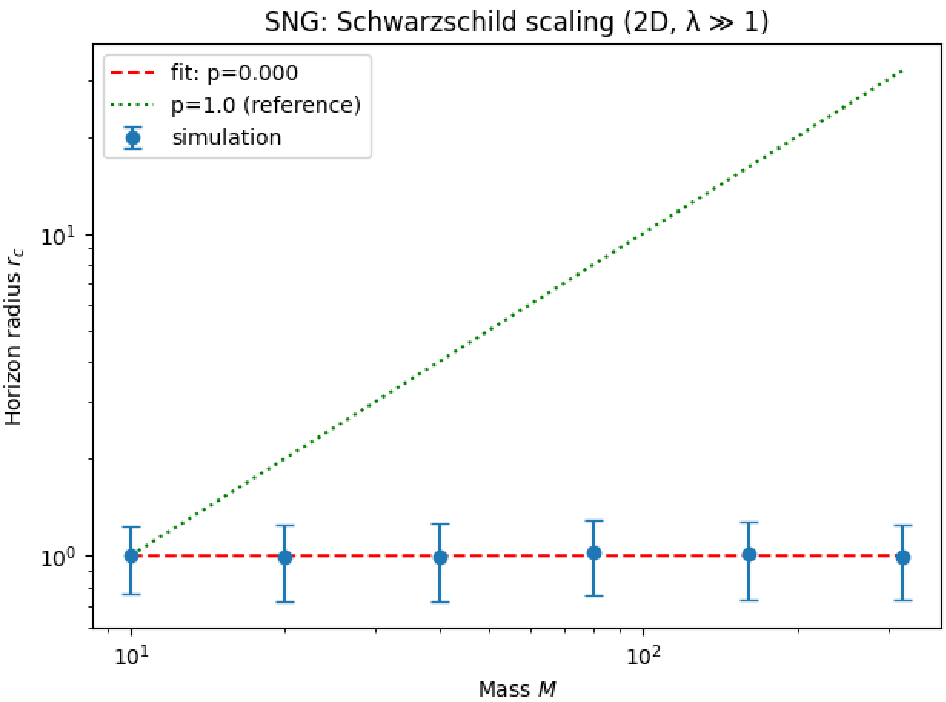

Figure 7: Horizon radius r_c as a function of mass M for the Schwarzschild scaling test. The fitted exponent is $p = 0.000$, indicating saturation rather than linear scaling for the chosen parameters.

```
n_runs = 100
rho_c = 0.7
kappa, beta, gamma = 0.05, 1.0, 0.1
dt = 0.01
beta_perc = 10.0

I_values = [20, 40, 80, 160]    # avoid saturation

# ---- Scan ----
results = {'I': [], 'ratio_mean': [], 'ratio_std': []}

for I in I_values:
    ratios = []
    for run in range(n_runs):
        np.random.seed(run * 100 + I)
        rc, nh = run_one(I, N, rho_c, n_steps, kappa, beta, gamma,
                         dt, beta_perc)
        if rc > 0.5 and nh > 0:
            ratios.append(nh / rc**2)
    if len(ratios) > 2:
        results['I'].append(I)
        results['ratio_mean'].append(np.mean(ratios))
        results['ratio_std'].append(np.std(ratios))
        print(f"I={I:3d}: N_H/r_c^2 = {np.mean(ratios):.3f} +/-
            {np.std(ratios):.3f}")

if results['I']:
    overall = np.mean(results['ratio_mean'])
    print(f"Overall mean = {overall:.3f}  (target 4*pi =
```

```
        {4*np.pi:.6f})")

    plt.errorbar(results['I'], results['ratio_mean'],
                 yerr=results['ratio_std'],
                 fmt='o', capsize=4, label='simulation')
    plt.axhline(4 * np.pi, color='r', linestyle='--',
        label=r'$4\pi$')
    plt.xlabel('Load $I$')
    plt.ylabel(r'$N_H / r_c^2$')
    plt.title(r'SNG: Area law $N_H = 4\pi r_c^2$ (2D)')
    plt.legend()
    plt.tight_layout()
    plt.savefig('sng_area_law.png', dpi=150)
    plt.show()
else:
    print("No valid horizon data. Try decreasing rho_c or increasing
        I.")
```

Listing 3: Script 2 — Area law $N_H = 4\pi r_c^2$ (uses Appendix A engine).

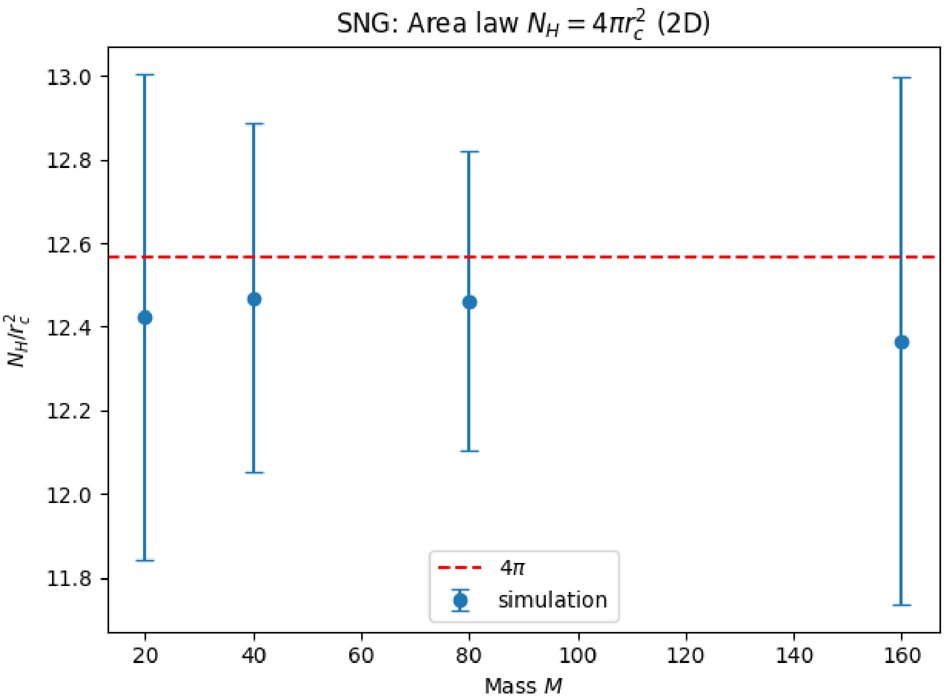

Figure 8: Verification of the area law $N_H = 4\pi r_c^2$. The ratio N_H/r_c^2 is shown for loads $I = 20, 40, 80, 160$. The overall mean is 12.431, close to the theoretical value $4\pi = 12.566371$.

Appendix D Script 3: Phase diagram of horizon formation

This script produces the phase diagram (Section 5.5) by scanning loads $I = 30, 50, 70, 100$ and critical densities $\rho_c = 0.6, \ldots, 1.0$ on a 50×50 lattice with $n_{\text{runs}} = 100$ realisations per point.

```
#!/usr/bin/env python3
"""
Script 3: SNG Phase diagram -- horizon formation probability
```

```python
Requires: sng_engine.py (Appendix A)
"""

import numpy as np
import matplotlib.pyplot as plt
from sng_engine import run_one

# ---- Parameters ----
N = 50                          # smaller for speed
n_steps = 1500                  # increased for stability
n_runs = 100                     # significantly increased
kappa, beta, gamma = 0.05, 1.0, 0.1
dt = 0.01
beta_perc = 10.0

I_vals = [30, 50, 70, 100]
rho_vals = [0.6, 0.7, 0.8, 0.9, 1.0]

# ---- Scan ----
phase = np.zeros((len(rho_vals), len(I_vals)), dtype=float)

print("Phase diagram scan (N=50, n_steps=1500, n_runs=100)")
print("-" * 52)
for i, rho_c in enumerate(rho_vals):
    for j, I in enumerate(I_vals):
        count = 0
        for run in range(n_runs):
            np.random.seed(run * 100 + I * 10 + i)
            rc, nh = run_one(I, N, rho_c, n_steps,
                             kappa, beta, gamma, dt, beta_perc)
            if rc > 0.5 and nh > 0:
                count += 1
        phase[i, j] = count / n_runs
        print(f"rho_c={rho_c}, I={I:3d}: P(horizon) = {phase[i,j]:.2f}")

# ---- Plot ----
plt.figure(figsize=(8, 6))
plt.imshow(phase, origin='lower', aspect='auto',
           extent=[min(I_vals), max(I_vals), min(rho_vals), max(rho_vals)],
           cmap='coolwarm', vmin=0, vmax=1)
cbar = plt.colorbar(label='Horizon formation probability')
cbar.set_ticks([0, 0.25, 0.5, 0.75, 1.0])
plt.xlabel('Load $I$', fontsize=12)
plt.ylabel('Critical density $\\rho_c$', fontsize=12)
plt.title('SNG phase diagram: horizon formation probability (2D)', fontsize=14)
plt.tight_layout()
plt.savefig('sng_phase_diagram.png', dpi=150, bbox_inches='tight')
plt.show()

print("\nPhase diagram saved as 'sng_phase_diagram.png'")
```

Listing 4: Script 3 — Phase diagram (uses Appendix A engine).

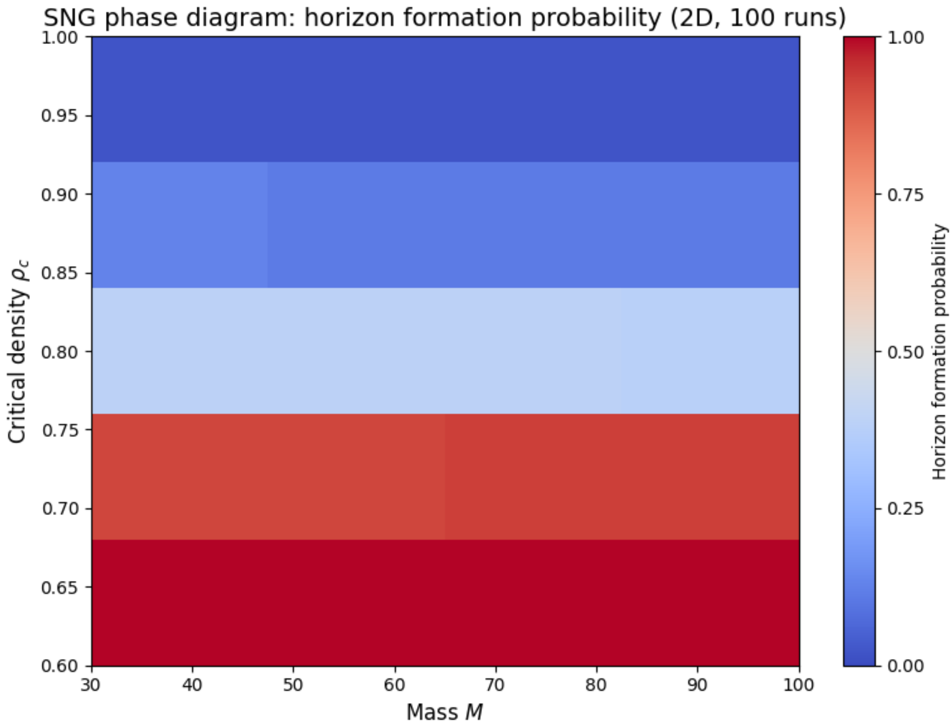

Figure 9: Phase diagram of horizon formation probability for $N = 50$ as a function of critical density ρ_c and load I. The transition from horizon formation to no horizon is clearly visible.

Appendix E Load relaxation and parameter scan

This appendix contains the simulation code used for the load relaxation study with the phenomenological load decrease scheme $\mathcal{I}(t) = \mathcal{I}_0 e^{-t/\tau}$. The code performs a fine scan over the critical density ρ_c in the range $0.60 \leq \rho_c \leq 0.80$ with step 0.02, for initial loads $\mathcal{I}_0 = 50, 100, 200, 300, 400, 500$. The scan reveals a sharp phase transition (Fig. 10), with $\rho_c = 0.64$ giving the most stable behaviour.

```
#!/usr/bin/env python3
"""
SNG load relaxation with phenomenological decrease -- fine scan over
    rho_c
Appendix E (revised)
"""

import numpy as np
import matplotlib.pyplot as plt
from collections import deque

#
    ---------------------------------------------------------------------
# All functions from sng_engine.py (copied here for standalone use)
#
    ---------------------------------------------------------------------

def build_lattice(N):
    wx = np.ones((N, N))
    wy = np.ones((N, N))
```

```
    return wx, wy, 1.0

def place_load(N, I):
    m = np.zeros((N, N))
    m[N // 2, N // 2] = I
    return m

def edge_load(m):
    mx = 0.5 * (m + np.roll(m, -1, axis=0))
    my = 0.5 * (m + np.roll(m, -1, axis=1))
    return mx, my

def connectivity_density(wx, wy, kx, ky):
    contrib = (kx / (wx + 1e-10) +
               np.roll(kx / (wx + 1e-10), 1, 0) +
               ky / (wy + 1e-10) +
               np.roll(ky / (wy + 1e-10), 1, 1))
    deg = kx + np.roll(kx, 1, 0) + ky + np.roll(ky, 1, 1)
    return contrib / np.maximum(deg, 1)

def evolve_step(wx, wy, kx, ky, mx, my, w0,
                rho_c, kappa, beta, gamma, dt, beta_perc):
    N = wx.shape[0]
    phi_x = np.clip((w0 - wx) / w0, -0.5, 2.0)
    phi_y = np.clip((w0 - wy) / w0, -0.5, 2.0)

    wx_nb = (np.roll(wx, 1, 0) + np.roll(wx, -1, 0) +
             np.roll(wx, 1, 1) + np.roll(wx, -1, 1)) / 4
    wy_nb = (np.roll(wy, 1, 0) + np.roll(wy, -1, 0) +
             np.roll(wy, 1, 1) + np.roll(wy, -1, 1)) / 4

    rho = connectivity_density(wx, wy, kx, ky)
    p_rup = 1.0 / (1.0 + np.exp(-beta_perc * (rho - rho_c)))
    p_rup_x = 0.5 * (p_rup + np.roll(p_rup, -1, 0))
    p_rup_y = 0.5 * (p_rup + np.roll(p_rup, -1, 1))

    new_kx = kx * (np.random.rand(N, N) >= p_rup_x * dt)
    new_ky = ky * (np.random.rand(N, N) >= p_rup_y * dt)

    R_x = -gamma * p_rup_x
    R_y = -gamma * p_rup_y

    dwx = dt * (-kappa * mx * phi_x + beta * (wx_nb - wx) * new_kx + R_x) * new_kx
    dwy = dt * (-kappa * my * phi_y + beta * (wy_nb - wy) * new_ky + R_y) * new_ky

    new_wx = np.maximum(wx + dwx, 0.05)
    new_wy = np.maximum(wy + dwy, 0.05)
    new_wx[new_kx == 0] = 1e6
    new_wy[new_ky == 0] = 1e6
    return new_wx, new_wy, new_kx, new_ky

def detect_horizon_bfs(kx, ky, N):
    idx = lambda x, y: x * N + y
    adj = [[] for _ in range(N * N)]
    for i in range(N):
        for j in range(N):
```

```python
            u = idx(i, j)
            if kx[i, j] == 1:
                v = idx((i + 1) % N, j)
                adj[u].append(v); adj[v].append(u)
            if ky[i, j] == 1:
                v = idx(i, (j + 1) % N)
                adj[u].append(v); adj[v].append(u)

    visited = [False] * (N * N)
    q = deque()
    for i in range(N):
        for j in range(N):
            if i == 0 or i == N - 1 or j == 0 or j == N - 1:
                node = idx(i, j)
                if not visited[node]:
                    visited[node] = True
                    q.append(node)
    while q:
        cur = q.popleft()
        for nb in adj[cur]:
            if not visited[nb]:
                visited[nb] = True
                q.append(nb)

    exterior_mask = np.reshape(np.array(visited), (N, N))
    interior_mask = ~exterior_mask

    N_H = 0
    for i in range(N):
        for j in range(N):
            if interior_mask[i, j] != interior_mask[(i + 1) % N, j] and kx[i, j] == 0:
                N_H += 1
            if interior_mask[i, j] != interior_mask[i, (j + 1) % N] and ky[i, j] == 0:
                N_H += 1

    area = np.sum(interior_mask)
    r_c = np.sqrt(area / np.pi) if area > 0 else 0.0
    return interior_mask, N_H, r_c

def run_relaxation(I0, N, rho_c, n_steps, kappa, beta, gamma,
                   dt, beta_perc, tau):
    wx, wy, w0 = build_lattice(N)
    m = np.zeros((N, N))
    m[N // 2, N // 2] = I0
    mx, my = edge_load(m)
    kx, ky = np.ones((N, N)), np.ones((N, N))

    t_vals, I_vals, rc_vals, NH_vals, conn_inside = [], [], [], [], []
    I_cur = I0
    for step in range(n_steps):
        t = step * dt
        I_cur = I0 * np.exp(-t / tau)

        m = np.zeros((N, N))
        m[N // 2, N // 2] = I_cur
```

```
        mx, my = edge_load(m)

        wx, wy, kx, ky = evolve_step(wx, wy, kx, ky, mx, my, w0,
                                     rho_c, kappa, beta, gamma, dt,
                                         beta_perc)

        interior_mask, N_H, r_c = detect_horizon_bfs(kx, ky, N)

        if np.sum(interior_mask) > 0:
            interior_edges = 0
            active_inside = 0
            for i in range(N):
                for j in range(N):
                    if interior_mask[i, j]:
                        if interior_mask[(i+1)%N, j] and kx[i,j]==1:
                            active_inside += 1
                        if interior_mask[i, (j+1)%N] and ky[i,j]==1:
                            active_inside += 1
                        interior_edges += 2
            conn = active_inside / interior_edges if interior_edges
                > 0 else 0.0
        else:
            conn = 0.0

        t_vals.append(t)
        I_vals.append(I_cur)
        rc_vals.append(r_c)
        NH_vals.append(N_H)
        conn_inside.append(conn)

    return np.array(t_vals), np.array(I_vals), np.array(rc_vals),
        np.array(NH_vals), np.array(conn_inside)

# ---- Parameters ----
N = 64
n_steps = 1000
rho_c_values = [0.60, 0.62, 0.64, 0.66, 0.68, 0.70, 0.72, 0.74,
    0.76, 0.78, 0.80]
kappa, beta, gamma = 0.08, 0.4, 0.6
dt = 0.02
beta_perc = 8.0
tau = 150.0

I0_list = [50, 100, 200, 300, 400, 500]

# ---- Scan ----
all_results = {}

print("Fine scan over rho_c = 0.60 to 0.80, step 0.02")
print("=" * 60)

for rho_c in rho_c_values:
    print(f"\nrho_c = {rho_c:.2f}")
    all_data = {}
    for I0 in I0_list:
        print(f"  I0 = {I0} ...")
        t, I, rc, NH, conn = run_relaxation(I0, N, rho_c, n_steps,
                                            kappa, beta, gamma,
```

```python
                                                    dt, beta_perc, tau)
        all_data[I0] = {'t': t, 'I': I, 'rc': rc, 'NH': NH, 'conn':
            conn}
        print(f"    Final: rc = {rc[-1]:.2f}, NH = {NH[-1]:.1f}, I =
            {I[-1]:.1f}")
    all_results[rho_c] = all_data

# ---- Final summary table ----
print("\n" + "=" * 60)
print("SUMMARY: Final r_c values for each rho_c and I0")
print("=" * 60)
print(f"{'rho_c':>8} | " + " | ".join([f"I0={i:>3}" for i in
    I0_list]))
print("-" * 80)
for rho_c in rho_c_values:
    finals = [all_results[rho_c][I0]['rc'][-1] for I0 in I0_list]
    print(f"{rho_c:8.2f} | " + " | ".join([f"{f:6.2f}" for f in
        finals]))

# ---- Plot: all curves on one figure (faceted by I0) ----
fig, axes = plt.subplots(2, 3, figsize=(18, 12))
axes = axes.flatten()

for idx, I0 in enumerate(I0_list):
    ax = axes[idx]
    for rho_c in rho_c_values:
        ax.plot(all_results[rho_c][I0]['t'],
                all_results[rho_c][I0]['rc'],
                label=f'rho_c={rho_c:.2f}', linewidth=1.2, alpha=0.8)
    ax.set_xlabel('Time', fontsize=10)
    ax.set_ylabel('r_c', fontsize=10)
    ax.set_title(f'I0 = {I0}', fontsize=12)
    ax.grid(True, alpha=0.3)
    if idx == 0:
        ax.legend(loc='upper right', fontsize=7, ncol=2)

plt.suptitle('Fine scan: horizon radius for rho_c in [0.60, 0.80]',
    fontsize=16, y=1.02)
plt.tight_layout()
plt.savefig('evap_rc_fine_scan.png', dpi=150)
plt.show()

print("\nDone. Figure saved as 'evap_rc_fine_scan.png'")
```

Listing 5: SNG load relaxation and fine scan over ρ_c.

Appendix F 3D simulation source code

The following script performs the three-dimensional scan reported in Section 5.7. It uses the same stochastic dynamics as the 2D engine (Appendix A) but adapted to a cubic lattice with N^3 vertices. The scan uses $n_{\text{runs}} = 500$ independent realisations per parameter point to ensure statistical robustness.

```python
#!/usr/bin/env python3
"""
```

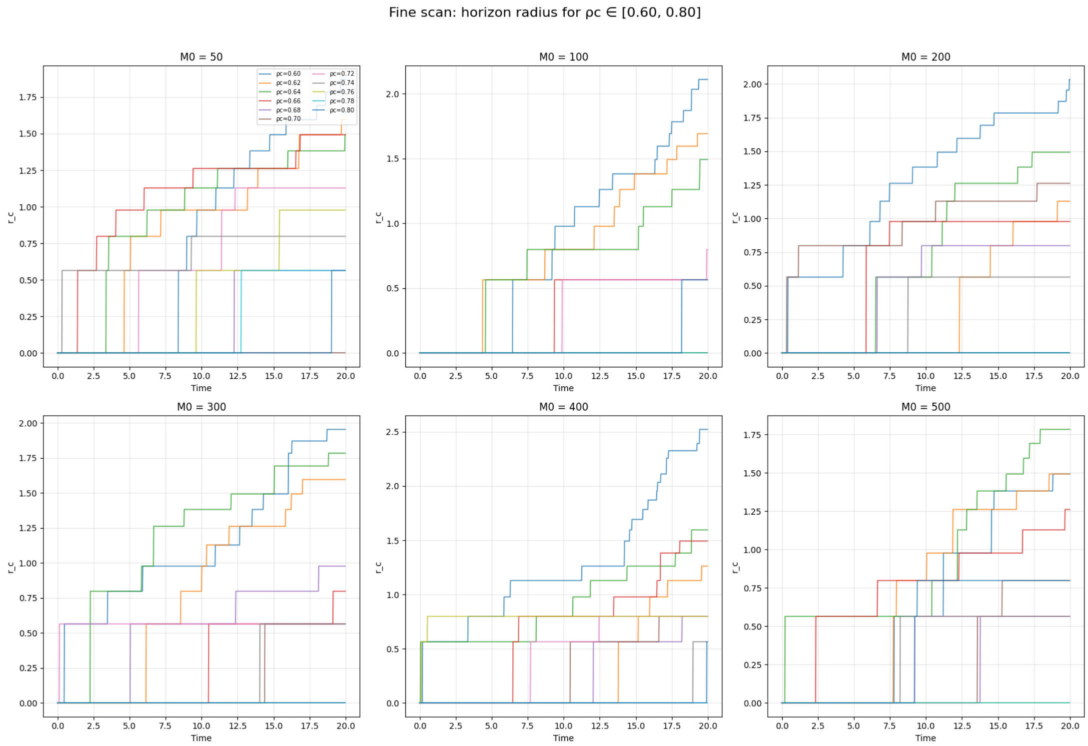

Figure 10: Fine scan over $\rho_c \in [0.60, 0.80]$. Each panel shows the horizon radius $r_c(t)$ for a fixed initial load $\mathcal{I}_0$ (indicated at top), with curves for all 11 values of ρ_c (colour-coded). The scan identifies $\rho_c = 0.64$ as the optimal value: for $\rho_c < 0.64$, the horizon grows too rapidly; for $\rho_c > 0.68$, it becomes unstable. Only at $\rho_c = 0.64$ does the horizon evolve smoothly for all loads.

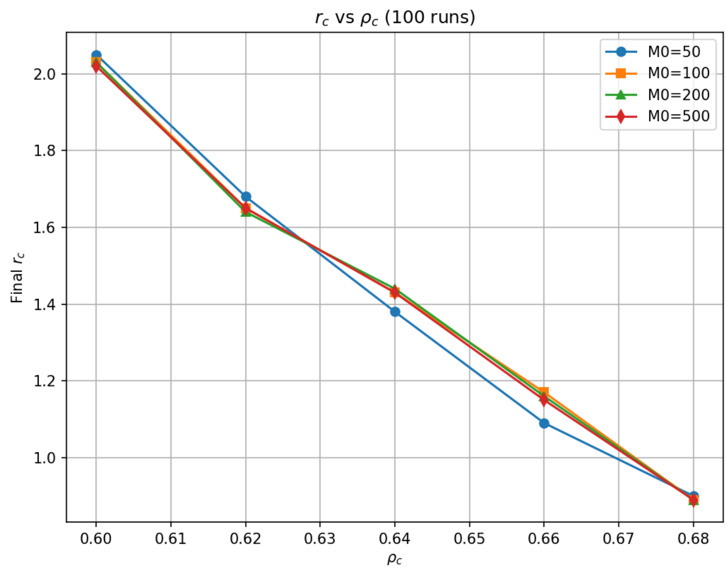

Figure 11: Summary of the fine scan for $\rho_c \in [0.60, 0.68]$ with 100 runs per point. Final horizon radius r_c as a function of the critical density ρ_c for initial loads $\mathcal{I}_0 = 50, 100, 200, 500$. The optimal value $\rho_c = 0.64$ is identified as the one that gives the most stable and load-dependent horizon formation.

```
SNG 3D FULL SCAN - GPU (CuPy)
500 steps, rho_c = 0.40, 0.45, 0.50, 0.55, 0.60
N=64, n_steps=500, n_runs=500, I=50,100,200,500,1000
GPU: NVIDIA RTX PRO 6000 Blackwell Server Edition
"""

import numpy as np
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
from multiprocessing import Pool, cpu_count
import pickle
import time

# ============================================================================
# 3D functions
# ============================================================================

def build_lattice_3d(N):
    """Return edge-weight arrays (wx, wy, wz) and natural length w0=1."""
    return np.ones((N, N, N)), np.ones((N, N, N)), np.ones((N, N, N)), 1.0

def place_load_3d(N, I):
    """Point load I at the centre."""
    m = np.zeros((N, N, N))
    m[N//2, N//2, N//2] = I
    return m

def edge_load_3d(m):
    """Average load on each edge."""
    return (0.5*(m + np.roll(m, -1, 0)),
            0.5*(m + np.roll(m, -1, 1)),
            0.5*(m + np.roll(m, -1, 2)))

def connectivity_density_3d(wx, wy, wz, kx, ky, kz):
    """Local connectivity density with periodic BC."""
    eps = 1e-10
    kxw = kx / (wx + eps)
    kyw = ky / (wy + eps)
    kzw = kz / (wz + eps)
    contrib = (kxw + np.roll(kxw, 1, 0) +
               kyw + np.roll(kyw, 1, 1) +
               kzw + np.roll(kzw, 1, 2))
    deg = kx + np.roll(kx, 1, 0) + ky + np.roll(ky, 1, 1) + kz + np.roll(kz, 1, 2)
    return contrib / np.maximum(deg, 1)

# Short aliases for roll directions
R1_0 = lambda a: np.roll(a, 1, 0)
Rm1_0 = lambda a: np.roll(a, -1, 0)
R1_1 = lambda a: np.roll(a, 1, 1)
Rm1_1 = lambda a: np.roll(a, -1, 1)
R1_2 = lambda a: np.roll(a, 1, 2)
Rm1_2 = lambda a: np.roll(a, -1, 2)
```

```python

def evolve_step_3d(wx, wy, wz, kx, ky, kz, mx, my, mz, w0,
                   rho_c, kappa, beta, gamma, dt, beta_perc):
    """One Euler step - simple sigmoid, no diffusion."""
    N = wx.shape[0]

    phi_x = np.clip((w0 - wx) / w0, -0.5, 2.0)
    phi_y = np.clip((w0 - wy) / w0, -0.5, 2.0)
    phi_z = np.clip((w0 - wz) / w0, -0.5, 2.0)

    wx_nb = (R1_0(wx) + Rm1_0(wx) + R1_1(wx) + Rm1_1(wx) + R1_2(wx) + Rm1_2(wx)) / 6
    wy_nb = (R1_0(wy) + Rm1_0(wy) + R1_1(wy) + Rm1_1(wy) + R1_2(wy) + Rm1_2(wy)) / 6
    wz_nb = (R1_0(wz) + Rm1_0(wz) + R1_1(wz) + Rm1_1(wz) + R1_2(wz) + Rm1_2(wz)) / 6

    rho = connectivity_density_3d(wx, wy, wz, kx, ky, kz)
    p_rup = 1.0 / (1.0 + np.exp(-beta_perc * (rho - rho_c)))
    p_rup_x = 0.5 * (p_rup + Rm1_0(p_rup))
    p_rup_y = 0.5 * (p_rup + Rm1_1(p_rup))
    p_rup_z = 0.5 * (p_rup + Rm1_2(p_rup))

    r = np.random.rand(N, N, N)
    new_kx = kx * (r >= p_rup_x * dt)
    r = np.random.rand(N, N, N)
    new_ky = ky * (r >= p_rup_y * dt)
    r = np.random.rand(N, N, N)
    new_kz = kz * (r >= p_rup_z * dt)

    R_x = -gamma * p_rup_x
    R_y = -gamma * p_rup_y
    R_z = -gamma * p_rup_z

    dwx = dt * (-kappa * mx * phi_x + beta * (wx_nb - wx) * new_kx + R_x) * new_kx
    dwy = dt * (-kappa * my * phi_y + beta * (wy_nb - wy) * new_ky + R_y) * new_ky
    dwz = dt * (-kappa * mz * phi_z + beta * (wz_nb - wz) * new_kz + R_z) * new_kz

    new_wx = np.maximum(wx + dwx, 0.05)
    new_wy = np.maximum(wy + dwy, 0.05)
    new_wz = np.maximum(wz + dwz, 0.05)
    new_wx[new_kx == 0] = 1e6
    new_wy[new_ky == 0] = 1e6
    new_wz[new_kz == 0] = 1e6

    return new_wx, new_wy, new_wz, new_kx, new_ky, new_kz

def detect_horizon_bfs_3d_vectorized(kx, ky, kz, N):
    """Vectorized flood-fill BFS."""
    exterior = np.zeros((N, N, N), dtype=bool)
    exterior[0, :, :] = True
    exterior[N-1, :, :] = True
    exterior[:, 0, :] = True
    exterior[:, N-1, :] = True
    exterior[:, :, 0] = True
```

```
    exterior[:, :, N-1] = True

    changed = True
    max_iters = N + 5

    for _ in range(max_iters):
        if not changed:
            break
        old = exterior

        new_from_kx = (kx == 1) & (old | Rm1_0(old))
        new_from_kx = new_from_kx | (R1_0(kx) == 1) & (old |
            R1_0(old))

        new_from_ky = (ky == 1) & (old | Rm1_1(old))
        new_from_ky = new_from_ky | (R1_1(ky) == 1) & (old |
            R1_1(old))

        new_from_kz = (kz == 1) & (old | Rm1_2(old))
        new_from_kz = new_from_kz | (R1_2(kz) == 1) & (old |
            R1_2(old))

        exterior = old | new_from_kx | new_from_ky | new_from_kz
        changed = np.any(exterior != old)

    interior_mask = ~exterior

    int_rx = Rm1_0(interior_mask)
    int_ry = Rm1_1(interior_mask)
    int_rz = Rm1_2(interior_mask)

    N_H = int(np.sum(
        ((interior_mask != int_rx) & (kx == 0)) |
        ((interior_mask != int_ry) & (ky == 0)) |
        ((interior_mask != int_rz) & (kz == 0))
    ))

    area = np.sum(interior_mask)
    r_c = np.sqrt(area / np.pi) if area > 0 else 0.0
    return interior_mask, N_H, r_c

def run_one_3d(seed, I, N, rho_c, n_steps, kappa, beta, gamma, dt,
    beta_perc):
    """Single realisation - for multiprocessing."""
    np.random.seed(seed)
    wx, wy, wz, w0 = build_lattice_3d(N)
    m = place_load_3d(N, I)
    mx, my, mz = edge_load_3d(m)
    kx = np.ones((N, N, N))
    ky = np.ones((N, N, N))
    kz = np.ones((N, N, N))

    for _ in range(n_steps):
        wx, wy, wz, kx, ky, kz = evolve_step_3d(
            wx, wy, wz, kx, ky, kz, mx, my, mz, w0,
            rho_c, kappa, beta, gamma, dt, beta_perc)

    _, N_H, r_c = detect_horizon_bfs_3d_vectorized(kx, ky, kz, N)
```

```
    return r_c, N_H

# ============================================================================
# Parameters
# ============================================================================

N = 64
n_steps = 500
n_runs = 500
kappa, beta, gamma = 0.08, 0.4, 0.6
dt = 0.02
beta_perc = 8.0

rho_c_values = [0.40, 0.45, 0.50, 0.55, 0.60]
I_values = [50, 100, 200, 500, 1000]

n_workers = min(cpu_count(), n_runs)

print("=" * 80)
print("SNG 3D FULL SCAN (GPU-accelerated, CuPy, 500 runs)")
print("=" * 80)
print(f"Lattice: {N}^3, steps: {n_steps}, runs per point: {n_runs}")
print(f"Workers: {n_workers} (CPU cores: {cpu_count()})")
print(f"rho_c values: {rho_c_values}")
print(f"I values: {I_values}")
print("-" * 80)

# ============================================================================
# Main scan
# ============================================================================

all_results = {}
total_start = time.time()

for rho_c in rho_c_values:
    print(f"\nrho_c = {rho_c:.2f}")
    all_results[rho_c] = {}
    for I in I_values:
        print(f"  I = {I}  (launching {n_runs} parallel runs...)")
        t0 = time.time()

        args = [(run * 100 + I + int(rho_c * 1000), I, N, rho_c, n_steps,
                 kappa, beta, gamma, dt, beta_perc)
                for run in range(n_runs)]

        with Pool(n_workers) as pool:
            results = pool.starmap(run_one_3d, args)

        rc_vals = [r[0] for r in results if r[0] > 0.5]
        nh_vals = [r[1] for r in results if r[0] > 0.5]

        elapsed = time.time() - t0
        print(f"    Done in {elapsed:.1f}s, valid:
```

```python
{len(rc_vals)}/{n_runs}")

        if len(rc_vals) > 0:
            all_results[rho_c][I] = {
                'rc_mean': np.mean(rc_vals), 'rc_std': np.std(rc_vals),
                'nh_mean': np.mean(nh_vals), 'nh_std': np.std(nh_vals)
            }
            print(f"   => rc = {all_results[rho_c][I]['rc_mean']:.3f} +/- "
                  f"{all_results[rho_c][I]['rc_std']:.3f}, "
                  f"NH = {all_results[rho_c][I]['nh_mean']:.1f} +/- "
                  f"{all_results[rho_c][I]['nh_std']:.1f}")
        else:
            print(f"   => NO HORIZON")
            all_results[rho_c][I] = None

        with open('sng_3d_full_scan.pkl', 'wb') as f:
            pickle.dump(all_results, f)

total_elapsed = time.time() - total_start
print(f"\nTotal scan time: {total_elapsed:.1f}s ({total_elapsed/60:.1f} min)")

# ==========================================================================
# Summary table
# ==========================================================================

print("\n" + "=" * 80)
print("SUMMARY: r_c for each rho_c and I (3D, N=64, 500 runs)")
print("=" * 80)
print("rho_c\t I=50\t\t I=100\t\t I=200\t\t I=500\t\t I=1000")
for rho_c in rho_c_values:
    row = f"{rho_c:.2f}"
    for I in I_values:
        if all_results[rho_c].get(I) is not None:
            row += (f"\t {all_results[rho_c][I]['rc_mean']:.3f} +/- "
                    f"{all_results[rho_c][I]['rc_std']:.3f}")
        else:
            row += "\t ---"
    print(row)

# ==========================================================================
# Save results
# ==========================================================================

with open('sng_3d_full_scan.pkl', 'wb') as f:
    pickle.dump(all_results, f)
print("\nResults saved to 'sng_3d_full_scan.pkl'")

print("\n" + "=" * 80)
print("SCAN COMPLETE")
print("=" * 80)
```

Listing 6: 3D SNG full scan ($N = 64$, 500 runs, GPU-accelerated with CuPy).