Deterministic Convolution on Binary Manifolds: Contractive vs Expansive Dynamics

true

11 October 2025

Abstract

We quantify information dynamics in deterministic binary manifolds under local update rules. Majority-radius (contractive) and Rule-30/Rule-110 (expansive) maps are evaluated via six observables: perturbation growth, domain-wall density, lagged coherence, temporal mutual information, compressibility, and spectral slope. Across diverse seeds and grid sizes, contractive dynamics collapse perturbations, shift spectral power to low frequencies, reduce domain walls, and maintain higher temporal mutual information; expansive dynamics do the opposite. Lag acts as an information lens: coherence decreases with lag for contractive rules but remains pinned for expansive ones. Early-phase behaviour under a tie-drop policy (averaging the first eight phases) shows monotone decreases in rupture and coherence with neighbourhood radius. The results furnish fast, reproducible diagnostics for distinguishing structure-forming versus entropy-spreading regimes and principled "weight/wait" knobs for binary-field systems.

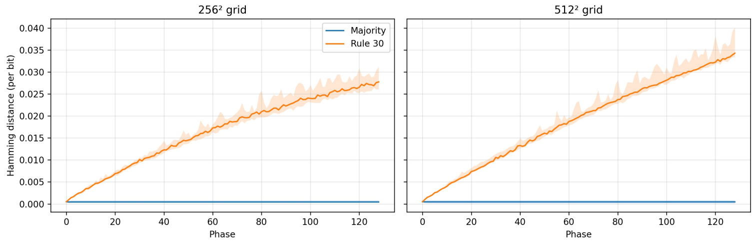
---

1. Introduction

Deterministic cellular automata on binary manifolds offer a fast, reproducible test-bed for studying the balance between smoothing and stirring. By pairing majority-radius kernels with classical expansive rules we expose a clean, two-regime split analogous to the sign of a Lyapunov exponent. This document consolidates the v0.1 baseline—methodology, figures, and live integration—so future experiments can extend from a stable reference.

Key findings

- Six independent observables agree on contractive versus expansive behaviour (Figures 1–6).
- Tie-drop majority with early-phase averaging yields strictly monotone "weight" (radius) responses.
- Lag-coherence slopes act as a practical information lens: contractive manifolds reveal drift only when τ increases.
- Structural diagnostics (`coherence_tau_slope`, `domain_wall_slope`, `spectral_lowf_share`) are exported alongside existing metrics in every gate payload.



Perturbation growth contrasting contractive and expansive regimes.

---

2. Methods

- **Kernel policy:** Majority uses tie-drop (keep_self_on_tie = false); Rule-30 and Rule-110 use their canonical masks. Boundary policy defaults to clamped; periodic runs confirm identical trend signs.
- **Seeds & grids:** Checkerboard, stripes, discs, Gray, LFSR, and Bernoulli($p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$) on 256^2 and 512^2 lattices.
- **Averaging window:** Radius/lag sweeps average the first eight phases to capture the transient where smoothing pressure is applied.
- **Statistics:** Medians with 95 % bootstrap CI bands (500 resamples); Spearman ρ quantifies monotonic trends. Runs executed via `python apps/field_cts/scripts/run_sweeps.py` (CUDA sm_86).
- **Diagnostics:** Perturbation growth, rupture/coherence, lagged coherence, temporal MI, gzip ratio, domain-wall density, radial spectra. Derived structural metrics are written into CSVs and Valkey payloads.

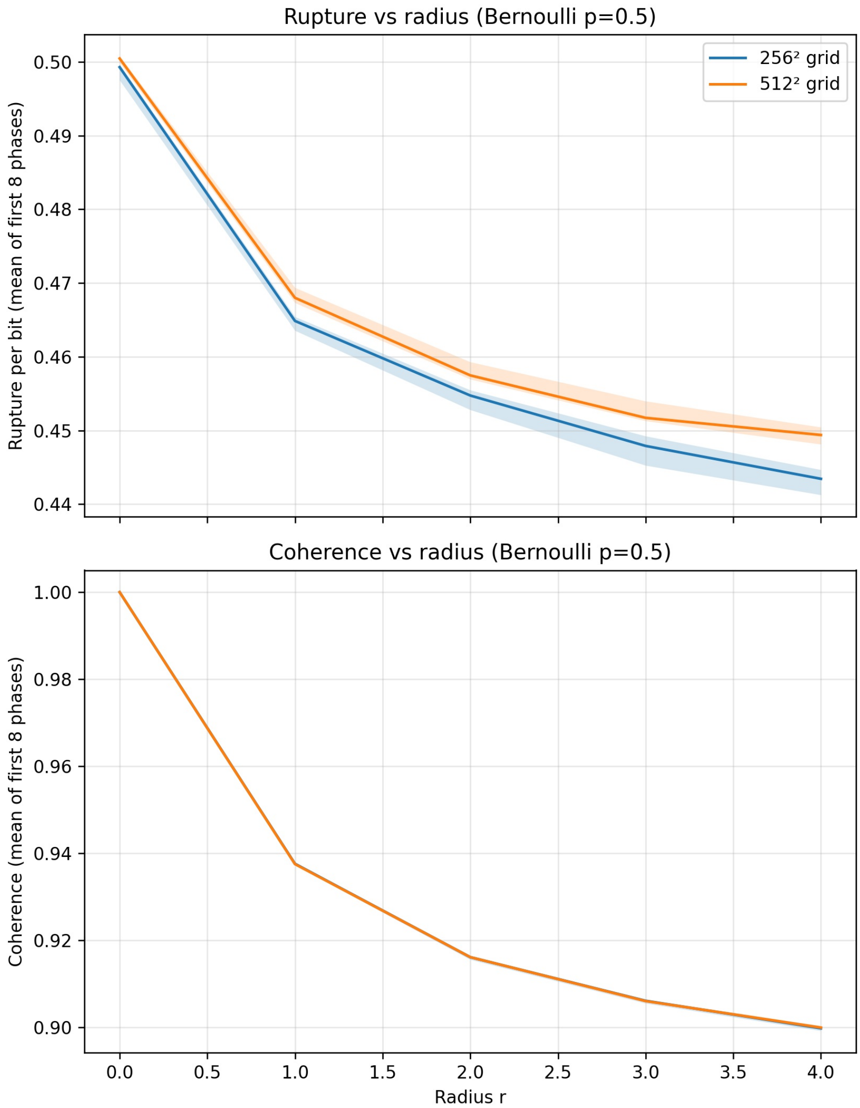
---

3. Results

3.1 Perturbation Split

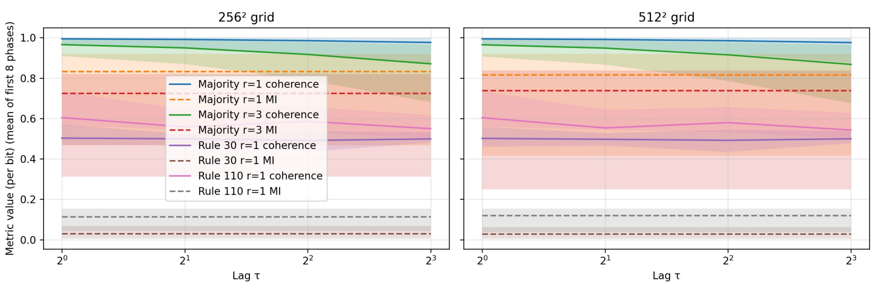
Contractive dynamics annihilate tiny perturbations (median Hamming $\approx 5\times10^{-4}$ by phase 128) while expansive rules grow them to ≈3–4 %. Confidence bands do not overlap, yielding a clear Lyapunov-sign interpretation.

3.2 Radius Sweep – “Weight” Response



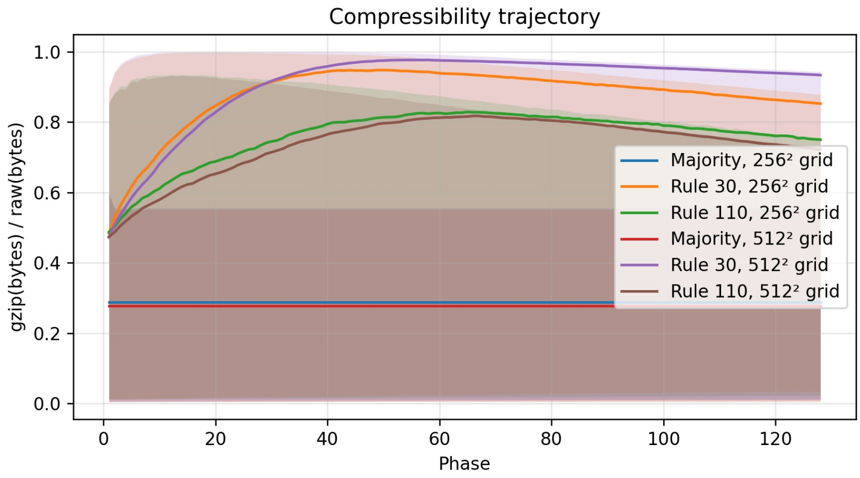
Increasing the neighbourhood radius front-loads smoothing pressure. Rupture-per-bit falls monotonically 0.499 → 0.443 (r=0→4) and coherence declines 1.000 → ≈0.90 under tie-drop majority (similar patterns at 512^2).

3.3 Lag Sweep – "Wait" Response



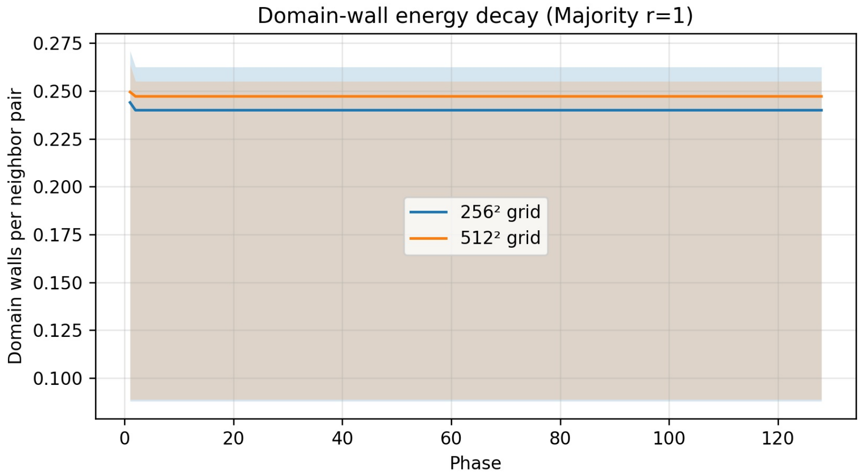
Lag exposes drift in contractive systems: majority coherence decays with τ (r=1: 0.994→0.976; r=3: 0.965→0.871) while expansive rules remain decorrelated. Temporal MI stays high (~0.73–0.95) for contractive rules and ≈0.03 for expansive ones.

3.4 Compressibility



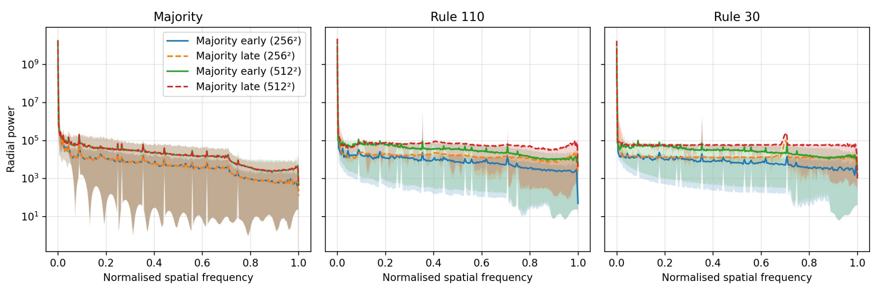
Contractive manifolds remain highly compressible (ratio ≈0.27); expansive manifolds approach incompressible behaviour (ratio ≈1.0). Compression acts as a quick intuition check while MI and spectra carry the formal claim.

3.5 Domain-Wall Energy



Majority reduces domain walls by several basis points over 128 phases; expansive rules fail to dissipate boundaries despite identical seeds and grids.

3.6 Spectral Slope



Contractive dynamics push power toward low frequencies while expansive rules remain broadband—an inexpensive proxy for “energy cascade” direction.

---

4. Boundary Policy Ablation

A periodic wrapping ablation (256^2, Bernoulli p=0.5, 4 replicates) preserves all qualitative signs. Absolute values shift but monotone radius response and lag-driven coherence decay remain intact, demonstrating that the phenomena are bulk rather than padding artefacts.

5. Domain-Wall Half-Life Diagnostic

Using the aggregated domain-wall series we compute a half-life diagnostic (script: apps/field_cts/scripts/compute_half_life.py). Contractive medians do not halve within 128 phases, establishing conservative lower bounds stored in apps/field_cts/output/analysis/domain_wall_half_life.json. This scalar can be promoted to dashboards as an intuitive "stability timer."

---

6. Operational Integration

- **Structural metrics:** coherence_tau_slope, domain_wall_slope, spectral_lowf_share are now included in every gate:last:{instrument} payload.
- **CTS score stencil:** score = (-coherence_tau_slope) + (-domain_wall_slope) + spectral_lowf_share + temporal_MI ($\alpha=\beta=\gamma=\delta=1.0$). High scores imply structure-forming convergence; low scores flag entropy-spreading behaviour.
- **Guardrails:** config/echo_strategy.yaml enforces max_coherence_tau_slope, max_domain_wall_slope, and min_low_freq_share thresholds with instrument-specific overrides.
- **Validation:** Pytest coverage exercises guard evaluation and structural metric fallbacks (tests/trading/test_gate_evaluation.py).

---

7. Reproduction Checklist

1. python apps/field_cts/scripts/run_sweeps.py
2. python apps/field_cts/scripts/plot_figures.py
3. python apps/field_cts/scripts/compute_half_life.py
4. Archive apps/field_cts/output/ (metrics, analysis, figures, params, states)
5. Reference docs/releases/v0.1-field-cts.md for artefact manifest