DOLICE

Causal Horizon Funeral Procession

葬　列

作品「Causal Sets / 葬列」を起点に、時間を流れではなく因果関係の部分順序として捉え直す——完全版 v10。投影規則 f の五公理を計算可能な形に落とし、中心命題「私たちは生まれてもいなければ滅してもおらず、ブラックホールの地平面の内側で葬列を成している」を順序論の事実として言い直す。

COMPLETE EDITION · v10 · JP

廣川政樹　Masaki Hirokawa

dolice.net

目次

	作品キャプション	3
	要旨	6
0	中心命題	7
1	投影規則 f（入出力仕様）	8
2	地平面・Controller・反例	12
3	Protocol Mapping（制作→提出→公開）	16
4	v9 —— 投影規則 f の最小公理系	17
5	v10 —— 地平面の内側	22
6	弦と膜 —— 超弦理論・M 理論からの読み替え	26
7	曼荼羅と生命の樹 —— 順序の同心投影	28
8	不生不滅 —— 涅槃の順序論的な言い直し	30
9	文化進化と多様性 —— A4 の時間発展	37
10	検証可能性と限界	39
11	結語 —— v10 が閉じるもの、開くもの	40
12	plate ギャラリー（v1〜v10）	41
	付録 A　コードの実行	53
	付録 B　用語集	54
	参考文献	55

作品キャプション



Causal Sets / 葬列（2023）—— 帽子で目を伏せた三つの像が、一輪の赤いバラを介して結ばれる。各々はセルの中で孤立しながらも、避けられない運命の絆によって一つの「列」を成す。

作品名 / Title	Causal Sets / 葬列
制作年 / Year	2023 年 4 月 / Apr 2023
サイズ / Dimensions	41 × 61 cm（縦 × 横 / H × W）
メディウム / Medium	デジタルメディア / Digital media
技法 / Technique	フォトコラージュ / Photo collage
作品ページ / Artwork page	https://dolice.net/art/causal-sets/

本作品「葬列」は、東洋の神秘思想と、量子力学と一般相対性理論をつなぐ概念的な架け橋である因果集合理論に着想を得ています。過去・現在・未来の出来事が一瞬のうちに展開されるという、原因と結果の同時性を探求し、時間の連続性という通念に挑戦しています。それぞれの存在は、自らのセルに孤立しているように描かれています。しかし、避けることのできない運命の絆によって互いにつながり、私たちの集合的な存在が形づくられていることが示唆されています。

また、「すべての個が唯一無二であり、不可欠な存在である」とのメッセージを伝えています。あらゆる存在が、世界の大きな変化に不可逆的に寄与しており、それぞれが宇宙の絶え間ない変容の中で重要な役割を担っているのです。星々の海を果てしなく続く比喩的な葬列を通じて、この作品は存在の儚さを浮かび上がらせます。そして、この移ろいやすい世界において、一瞬一瞬の出来事と、出会いというご縁を大切にすることの意義を私たちに問いかけてきます。

— 関連作（補助的参照）

「元型」の反復・入れ子の発想は、本稿における rank の入れ子（v5）と、§7 の曼荼羅投影——同心の環として現れる段——の読解に、補助的な参照枠を与える。



Archetypus / 元型（2020）—— 鳥の嘴の仮面を反復複製した入れ子の像。合わせ鏡のような終わりのない反復。

作品名 / Title	Archetypus / 元型
制作年 / Year	2020 年 7 月 / Jul 2020
サイズ / Dimensions	123 × 99 cm（縦 × 横 / H × W）
メディウム / Medium	デジタルメディア / Digital media
技法 / Technique	フォトコラージュ / Photo collage
作品ページ / Artwork page	https://dolice.net/art/archetypus/

この作品では「原型への回帰と反復」というテーマに沿って、鳥の仮面を被った女性を繰り返し複製しています。これは人間の持つ熾烈な願望を表現しています。狂気を支える論理が芸術作品、ひいては人類の文明には必要不可欠であり、作家にとって論理は感覚を抑える手段に過ぎないため、両者の共存が合わせ鏡のような反復を生むことになります。そこには終焉がなく、ただ入れ子構造になった世界がループしているのみです。人類の歴史や文化、戦争、紛争、そして宇宙もこのようにして構築されてきたのではないでしょうか。

要旨

本稿は、作品「葬列」をブラックホール時空の地平面 B の近傍に固定された構造として捉え、時間を「流れ」ではなく「因果関係の部分順序」として扱う。v1〜v9 で経験的に積み上げられた投影規則 f は、v9 において五つの最小公理（A1〜A5）から再構成された。完全版 v10 である本稿は、その公理系を実際に計算できる形――散布された因果集合・ヌル面としての地平面・四つのパラメータ $\theta = (p, q, \sigma, \kappa)$――に落とし、安定領域 Ω_stable を相図として求め、A5 の上界 κ_max を最大流（bit threads）から数値的に導く。さらに、理論の中心命題「私たちは生まれてもいなければ滅してもおらず、ブラックホールの事象の地平面の内側にいて、すべての人が葬列を成している」を、順序論の三つの定理的事実――窓の縁の定理（最小元・最大元は観測窓の産物）、地平面上の固有時 $\tau = 0$、Dilworth の鎖分解――として言い直す。超弦理論・M 理論（T 双対・ブレーン上の世界）、曼荼羅と生命の樹（順序の同心投影）、涅槃（不生不滅）との対応は、どこまでが定理でどこからが比喩かを明示したうえで述べる。すべてのコードは一行ごとに注釈を付し、図版はそのコードから再生成した。

キーワード　因果集合理論・事象の地平面・ホログラフィック原理・ビット・スレッド・Dilworth の定理・最小公理系・相図・T 双対・曼荼羅・生命の樹・不生不滅・文化進化

第 0 章

中心命題

本稿が最終的に述べたいことは三つの文に収まる。以下、各命題に対して「順序論の事実」「物理の対応」「東洋思想の対応」を分けて扱い、どの層で何が示されるかを明示する。命題を信じてもらうためではなく、命題がどの程度まで計算可能かを測るためである。

P 1

私たちは生まれてもいなければ、滅してもいない。

順序論の事実：有限の観測窓の中では「過去を持たない要素」（始まり）と「未来を持たない要素」（終わり）が必ず現れるが、窓を固定したまま因果集合を広げると、その割合は 0 に向かう（§8.1・図 8）。始まりと終わりは順序の性質ではなく、窓の縁の性質である。物理：地平面上では固有時が流れない（$\tau = 0$・§8.2）。東洋思想：般若心経の「不生不滅」。

P 2

この世界は、ブラックホールの事象の地平面の内側にある。

順序論の事実：ヌル面 u = u_h で因果集合を二分すると、糸は外側から内側へしか越えられない（§2・§5.1）。内側から見ると地平面は「越えられない過去」であり、外側から見ると「読み取るしかない境界」である。物理：ホログラフィック原理——境界のデータから bulk が復元される（§5.2）。本稿が扱う地平面は、A1〜A5 が同時に成立する Ω_stable の内部にある地平面——静的なシュヴァルツシルト地平面の理想化——である。東洋思想：阿頼耶識（蔵識）から現象世界が展開する（§4.6）。

P 3

すべての人々は、葬列を成している。

順序論の事実：Dilworth の定理。任意の有限な部分順序は、その幅に等しい本数の鎖（全順序な列）に分割でき、すべての要素はちょうど一つの鎖に属する（§8.4・図 10）。列の外に立つ者はいない。物理：ビット・スレッド——地平面を越える独立な糸の最大本数は最小カットに等しい（§5.2）。東洋思想：縁起。要素は関係の束であって、関係の外に実体を持たない（無我）。

第 1 章

投影規則 f（入出力仕様）

f は Model／Controller の状態を View 出力へ写像する最小規則である。「最小」とは表現を縛ることではなく、差分が追跡可能な範囲で自由度を確保するという意味である。同じ入力なら同じ出力が得られ、出力の違いがどの要因に帰属するかが追える。

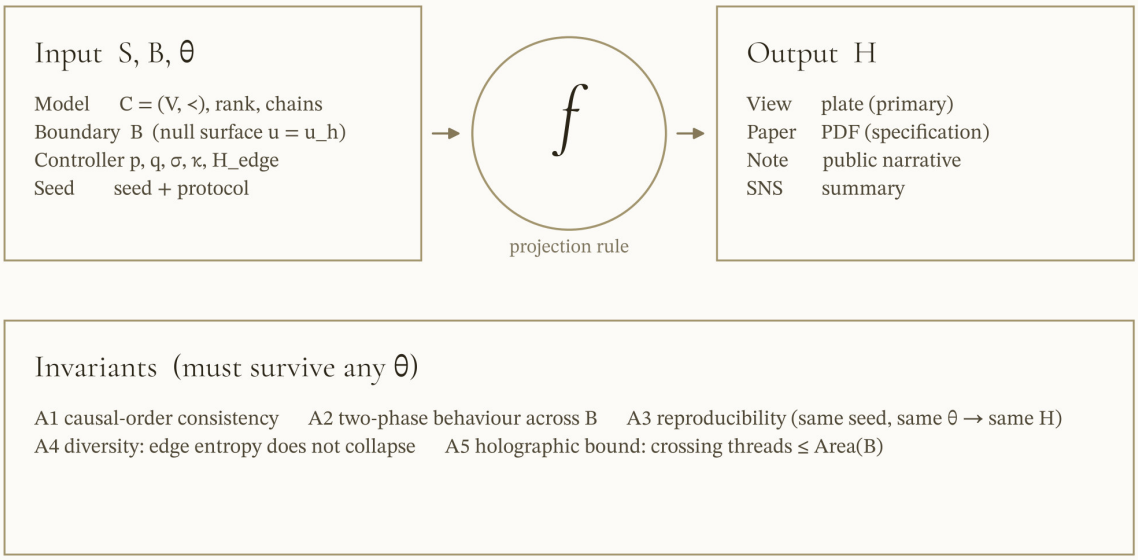

図 1 投影規則 f の入出力と不変量。

— 1.1 内部状態（最小定義）

```
Input model : S = (V, ≺), rank, chains
Boundary    : B  (null surface u = u_h)
Controller  : θ = (p, q, σ, κ), H_edge
Seed        : seed + protocol
Output      : H = f(S; B, θ)
```

v10 では、Model を「光円錐座標に一様散布された因果集合」として具体化する。散布（sprinkling）は因果集合理論の標準的な手続きであり、時空の体積に比例した個数の点をポアソン過程で落とす。順序 ≺ は光円錐で定まる。これ以外の構造——距離・時計・座標——を要素に持たせない。順序と個数だけが幾何を与える、という因果集合理論の原則（“order + number = geometry”）をそのまま採る。

```
共通の準備
# 数値計算の土台。点群・距離・乱数のすべてをこのライブラリで扱う
import numpy as np
# 図版の描画に使う。本稿の図はすべてこのモジュールで生成する
import matplotlib.pyplot as plt
# 複数パネルを一枚の plate に並べるためのレイアウト
from matplotlib.gridspec import GridSpec
# 最大流・最小カット（bit threads）の計算に使うグラフ理論ライブラリ
import networkx as nx
# 自然対数と円周率（エントロピーと角度の計算に必要）
import math
# Controller θ を「名前付きの値の束」として宣言するための装飾子
from dataclasses import dataclass
```

```
因果集合の散布（SPRINKLING）
def sprinkle_diamond(n=320, seed=7):
    # 再現可能性のため、乱数は必ず seed から生成する（A3 の前提）
    rng = np.random.default_rng(seed)
    # 光円錐座標 u, v を [0, 1) に一様散布する（ポアソン散布の近似）
    u = rng.random(n)
    # v も同様に。u と v が独立なら、ダイヤモンド領域に一様に落ちる
    v = rng.random(n)
    # 時間座標 t = (u + v) / √2。上へ行くほど「未来」
    t = (u + v) / math.sqrt(2.0)
    # 空間座標 x = (v − u) / √2。左右の広がり
    x = (v - u) / math.sqrt(2.0)
    # 時間順に並べ替えておくと、因果行列が上三角になり扱いやすい
    order = np.argsort(t)
    # (t, x) の列を返す。これが因果集合の「点」V である
    return np.column_stack([t[order], x[order]])
```

```
def causal_matrix(pts):
    # 要素数 n。行列は n × n になる
    n = len(pts)
    # すべての時間差 Δt = t_j − t_i を一度に計算する
    dt = pts[None, :, 0] - pts[:, None, 0]
    # すべての空間差 Δx = x_j − x_i も同様
    dx = pts[None, :, 1] - pts[:, None, 1]
    # i ≺ j ⇔ j が i の未来光円錐の内側にある（Δt > |Δx|）
    prec = (dt > np.abs(dx))
    # 自分自身とは因果関係を結ばない（非反射律）
    np.fill_diagonal(prec, False)
    # 真偽の行列 C[i, j] = (i ≺ j)。これが時間の「部分順序」そのもの
    return prec
```

```
def rank_longest_chain(C):
    # 要素数
    n = C.shape[0]
    # 各要素の rank（過去へ向かう最長鎖の長さ）を 0 で初期化
    rank = np.zeros(n, dtype=int)
    # 時間順に走査する（C は上三角なので過去はすべて小さい添字）
    for j in range(n):
        # j の過去にある要素の添字をすべて集める
        past = np.where(C[:, j])[0]
        # 過去が空なら rank は 0（「生まれ」のない要素）
        if len(past):
            # 過去の最大 rank + 1 が自分の rank。これが「葬列の何段目か」
            rank[j] = rank[past].max() + 1
    # 全要素の rank を返す
    return rank
```

```
def links(C):
    # 推移閉包を除いた「リンク」（直接の因果）だけを残す。C² が真なら中継がある
    via = (C.astype(int) @ C.astype(int)) > 0
    # 中継のない関係だけがリンク。ハッセ図の辺にあたる
    return C & ~via
```

最長鎖の長さを rank と呼ぶ。これが「葬列の何段目か」である。

— 1.2　次元の創発

「要素には順序しかない」という原則の試金石として、次元を順序だけから取り出す。Myrheim と Meyer は、時空次元 d の因果集合では関係のある対の割合 f が d だけで決まる関数 f(d) = Γ(d+1)Γ(d/2) / (2Γ(3d/2)) になることを示した（d = 2 で 1/2、d = 4 で 1/10）。観測した f から逆に d を解くと、本稿の散布（1+1 次元）では d ≈ 2.01 が得られる（図 12-I）。要素は座標を持たないのに、次元が出てくる。時空の形は要素の属性ではなく、関係の統計である。

```
次元の創発（MYRHEIM–MEYER 推定）
def ordering_fraction(C):
    # 要素数
    n = C.shape[0]
    # 関係のある対の数 ÷ 全対の数。「順序の密度」
    return float(C.sum()) / (n * (n - 1) / 2.0)
```

```
def myrheim_meyer_dimension(C):

# 時空次元 d の因果集合では、関係のある対の割合 f は f(d) = Γ(d+1)Γ(d/2) / (2 Γ(3d/2)) になる（Myrheim–Meyer。d = 2 で 1/2、d = 4 で 1/10）
    f = ordering_fraction(C)
    # 公式を d の関数として書く
    def expected(d):
        # ガンマ関数で計算する
        return math.gamma(d + 1) * math.gamma(d / 2) / (2 * math.gamma(1.5 * d))
    # d を 1 から 6 まで二分探索し、expected(d) = f となる d を求める（f は d について単調減少）
    lo, hi = 1.0, 6.0
    # 40 回で十分な精度
    for _ in range(40):
        # 中点
        mid = 0.5 * (lo + hi)
        # 中点の期待値が観測より大きければ、次元はもっと大きい
        lo, hi = (mid, hi) if expected(mid) > f else (lo, mid)
    # 順序だけから推定した次元。要素は座標を持たないのに、次元が出てくる
    return 0.5 * (lo + hi)
```

— 1.3　出力体系

plate／paper PDF／note draft／SNS text。plate を一次成果として固定し、他の出力は同一の Model／Controller 状態（同一 seed とパラメータ集合）から派生させる。v10 の plate（図 11）は四面から成る：I 散布された因果集合と地平面、II ホログラム H、III 地平面で分けた二点相関、IV rank の分布。

第 2 章

地平面・Controller・反例

不変量とは「壊すと作品として成立しない骨格」である。本稿では少なくとも、因果整合、地平の二相性、層構造（rank・chains）、多様性（entropy の非崩壊）を不変量として固定する。反例（counterexamples）を明示することで禁止領域が定まり、安定条件の意味が具体化する。

— 2.1　地平面をヌル面として置く

地平面 B を、光円錐座標の一方 u = (t − x)/√2 が一定となる面として置く。これはクルスカル座標における U = 0、すなわちシュヴァルツシルト地平面の局所的な形である。因果関係は u を減らすことができないので、糸は外側から内側へ越えることはできても、内側から外側へ戻ることはできない。「内側」とは、地平面を過去として持つ領域のことである。

```
地平面による二分
def horizon_split(pts, u_h=0.5):
    # 光円錐座標 u = (t − x)/√2 を復元する。地平面はヌル面 u = u_h（クルスカル座標の U = 0 に相当）
    u = (pts[:, 0] - pts[:, 1]) / math.sqrt(2.0)
    # u ≥ u_h の要素が地平面の「内側」。因果関係は u を減らせないので、内側から外側へ戻る糸は存在しない
    inside = u ⩾ u_h
    # 内側の真偽ベクトルを返す。外側はその否定
    return inside
```

```
def cross_links(L, inside):
    # リンクの始点と終点の添字をすべて取り出す
    i, j = np.where(L)
    # 始点と終点で inside が異なるものが「地平面を越える糸」
    crossing = inside[i] ≠ inside[j]
    # 越える糸の本数と、その添字の組を返す
    return int(crossing.sum()), list(zip(i[crossing], j[crossing]))
```

```
def horizon_area(L, inside):
    # 地平面の「面積」を、地平面を横切るリンクの本数で測る（Dou–Sorkin の地平面エントロピー：面積 ∝ 横切るリンク数）
    return cross_links(L, inside)[0]
```

地平面の「面積」は Dou–Sorkin に倣い、地平面を横切るリンクの本数で測る。

— 2.2　Controller θ と f の実装

θ の四つの成分は、v1〜v9 を通じて経験的に意味が固まった。p は地平面を越える糸の透過率、q は地平面を越えない糸の保持率、σ は配置の揺らぎ、κ は柱を横切って無作為に足される干渉の割合である。f はこれらを順に適用する——揺らぎ、間引き、干渉——だけの写像である。

CONTROLLER Θ と投影規則 F

```
@dataclass
class Controller:
    # p: 透過／混合率。地平面を越える糸がどれだけ通るか (0, 1)
    p: float
    # q: 選択／閉鎖率。地平面を越えない糸をどれだけ残すか (0, 1)。q → 0 で孤立化
    q: float
    # σ: 揺らぎ。配置に加えるジッターの標準偏差 (≥ 0)
    sigma: float
    # κ: 跨柱干渉。柱（pillar）を横切る接続の割合 (≥ 0)
    kappa: float
```

```
def project(pts, L, inside, theta, seed=0):
    # 投影にも乱数が要る（間引きと揺らぎ）。seed を固定して再現可能にする
    rng = np.random.default_rng(seed)
    # 出力 H の座標 = 入力座標 + ガウス揺らぎ σ（A1 の試練）
    H = pts + rng.normal(0.0, theta.sigma, size=pts.shape)
    # リンクの両端を列挙する
    i, j = np.where(L)
    # 地平面を越える糸は確率 p で残し、それ以外は確率 q で残す
    keep_p = np.where(inside[i] != inside[j], theta.p, theta.q)
    # 一様乱数と比較して、残す糸を決める
    keep = rng.random(len(i)) < keep_p
    # 残った糸を (i, j) の配列にする
    E = np.column_stack([i[keep], j[keep]])
    # κ の割合だけ、遠く離れた要素どうしを無作為に結ぶ（跨柱干渉）
    m = int(theta.kappa * len(E))
    # 無作為な組を生成する（添字は時間順なので a < b にして向きを揃える）
    extra = np.sort(rng.integers(0, len(pts), size=(m, 2)), axis=1)
    # 自己ループを除く
    extra = extra[extra[:, 0] != extra[:, 1]]
    # 元の糸と干渉の糸を合わせたものが、ホログラム H の辺集合
    return H, np.vstack([E, extra]) if m else E
```

— 2.3　観測量

公理を検査するには、H から読める量が要る。辺のエントロピー（次数分布のシャノンエントロピー）、到達可能性の深さ（最長経路）、地平面を越える糸の本数、二点相関 g(r)。g(r) は同じ領域への一様散布を分母にとる。散布そのものには尺度がない——相関は位置ではなく順序から来る——ことが、図 11 の III で読める。

```
観測量
def edge_entropy(E, n):
    # 各要素の次数（結ばれている糸の本数）を数える
    deg = np.bincount(E.ravel(), minlength=n).astype(float)
    # 次数を確率分布に正規化する（糸が一本もなければ 0 を返す）
    if deg.sum() == 0:
        # 孤立化：多様性ゼロ
        return 0.0
    # 確率 p_i = deg_i / Σdeg
    P = deg / deg.sum()
    # ゼロの項を除いてシャノンエントロピー −Σ p log p を計算する
    P = P[P > 0]
    # 自然対数で返す。最大値は log n（完全な均質化）
    return float(-(P * np.log(P)).sum())
```

```
def reachability_depth(E, n):
    # 糸を有向辺とするグラフを作る（時間順の添字なので自動的に非巡回）
    G = nx.DiGraph()
    # 全要素を頂点として登録する
    G.add_nodes_from(range(n))
    # 糸を辺として登録する
    G.add_edges_from(map(tuple, E))
    # 最長経路の長さ = 到達可能性の深さ。葬列が「どこまで続くか」
    return nx.dag_longest_path_length(G)
```

```
def pair_correlation(H, ref, r_max=0.6, bins=36):
    # すべての点対の距離を計算する（上三角のみ・自分自身との距離 0 は除く）
    d = np.sqrt(((H[:, None, :] - H[None, :, :]) ** 2).sum(axis=2))[np.triu_indices(len(H), k=1)]
    # 参照点群（同じ領域への一様散布）の対距離も同様に
    d0 = np.sqrt(((ref[:, None, :] - ref[None, :, :]) ** 2).sum(axis=2))[np.triu_indices(len(ref),
k=1)]
    # 距離のヒストグラム（r ごとの対の数）
    hist, edges = np.histogram(d, bins=bins, range=(0.0, r_max))
    # 参照のヒストグラム。領域の形による偏りを打ち消す分母
    hist0, _ = np.histogram(d0, bins=bins, range=(0.0, r_max))
    # ビンの中心を r とする
    r = 0.5 * (edges[:-1] + edges[1:])
    # g(r) = 観測 / 参照（対の総数で正規化）。g ≈ 1 が無相関、ピークが構造
    g = (hist / max(hist.sum(), 1)) / np.maximum(hist0 / max(hist0.sum(), 1), 1e-9)
    # r と g(r) を返す
    return r, g
```

— 2.4　反例

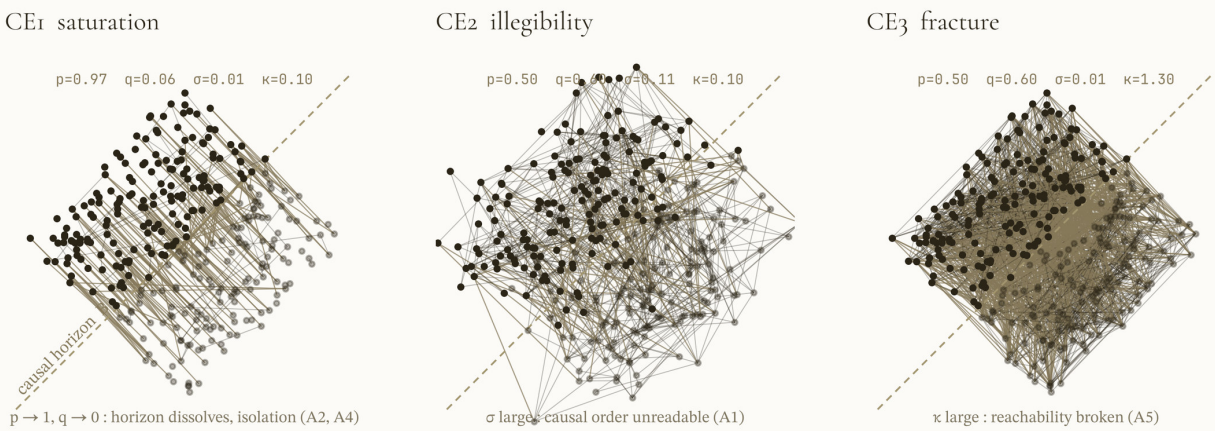

図 2　反例（崩壊モード）。CE1 飽和（p→1, q→0）、CE2 判読不能（σ 大）、CE3 断裂（κ 大）。v10 のモデルで実際に計算した。

三つの崩壊は、図 2 の三面にそのまま現れる。CE1 では糸が地平面を自由に越え、内側の糸は消え、二相性が失われる。CE2 では揺らぎがリンクの最小長を超え、どの要素がどの要素の後にあるかが読めない。CE3 では地平面を越える糸が地平面の面積を超え、列が断裂する。

第 3 章

Protocol Mapping（制作→提出→公開）

View は plate／論文 PDF／note 原稿／SNS text の 4 出力を基準とする。plate は制作の一次成果、論文は仕様の固定、note は公開のための語り、SNS text は告知・共有のための要約である。同一の内部状態から 4 出力を派生させることで、制作工程は追跡可能になり、作品と言語（仕様）が乖離しない。v10 ではさらに、図版そのものを本稿のコードから生成することで、図と仕様の乖離も閉じた。

第 4 章

v9 —— 投影規則 f の最小公理系

v1〜v8.1 を通じて経験的に固定されてきた投影規則 f を、最小の公理系から演繹的に再構成する。提案する公理は五つ（A1〜A5）である。これらが同時に成立するとき、かつそのときに限り、f は「葬列」という構造——孤立したセルが因果の糸で結ばれ、列を成す——を生成できる。各崩壊反例（CE1〜CE3）は特定の公理の違反として再解釈され、安定領域は「全五公理が同時に成立するパラメータの集合」として代数的に記述される。

4.1 既知の構造の整理

```
H = f(S; B, θ)
S = (V, ≺)                  causal set: elements V with causal order ≺
B                           causal horizon (null surface u = u_h)
θ = { p, q, σ, κ, H_edge }  Controller parameters
H                           View = Hologram (output)
```

4.2 五つの最小公理

A1 順序保存

```
a ≺ b in S   ⇒   f(a) precedes f(b) in H
```

f は S の偏順序を H の中に保存しなければならない。時間が「流れ」ではなく「到達可能性の深さ」であるなら、f はその深さの序列を壊してはならない。

違反するとき：σ が大きすぎる（CE2）。揺らぎがリンクの最小長 ≈ 1/√n を超え、どのセルがどのセルの「後」かが読めなくなる。

A2 地平面の二相性

```
B splits V into V_inside and V_outside;
f(V_inside) and f(V_outside) are structurally distinct.
```

p → 1 のとき内外が混ざり、地平面が消える。地平面がなければ葬列は「列」にならない——単なる点群である。逆に越える糸が途絶えれば、二つの世界は互いに無関係になり、やはり列は成らない。

違反するとき：p→1, q→0（CE1）。飽和は、地平面の意味喪失と多様性の崩壊が同時に起きる状態である。

A3　局所連続性

```
for small δθ, the change δH is bounded and continuous;
θ lies in the interior of Ω_stable.
```

これは「制作の再現可能性」の公理である。同じ seed と同じ θ から同じ H が得られる——それが Protocol Mapping の基盤だ。安定領域の境界では連続性が壊れ、相転移的な崩壊が起きる。v10 では θ の四近傍で他の公理が保たれることを検査する。

違反するとき：CE1〜CE3 のすべての境界。安定領域の外縁が相転移点である。

A4　エントロピー非崩壊

```
H_edge( f(S; B, θ) ) ≥ H_min > 0
```

辺のエントロピーが 0 に向かうとき、f は多様性を失う。全接続が等確率になれば（均質化）、あるいは全接続がゼロになれば（孤立化）、葬列は意味を持たない——「運命の絆」が消えるからだ。v7 で edge entropy が導入されたのは、この公理を可視化するためだった。

違反するとき：CE1（q→0 による孤立化）。

A5　ホログラフィック境界制約

```
| cross-horizon threads |  ≤  Area(B) / ℓ_P²
```

地平面を越える因果の糸の数は、地平面の「面積」に比例した上界を持つ。これは Bekenstein–Hawking 境界エントロピーの離散アナログである。v10 では面積を Dou–Sorkin に倣って「地平面を横切るリンクの本数」で測り、上界を最大流で検査する（§5.2）。

違反するとき：κ が大きすぎる（CE3）。越える糸が多すぎると到達可能性が破れ、列が断裂する。

4.3　公理と崩壊反例の対応

崩壊反例	条件	違反する公理	現象
CE1	p→1, q→0	A2, A4	地平面消失 ＋ 孤立化
CE2	σ 大	A1	因果順序の判読不能
CE3	κ 大	A5	到達可能性の断裂

— 4.4　安定領域の代数的記述と相図

```
Ω_stable = { θ = (p, q, σ, κ) :
   A1:  σ < σ_max(n)          σ below the minimal link length
   A2:  r_min < ρ(θ) < r_max  crossing ratio ρ neither severed nor saturated
   A3:  θ ∈ int(Ω_stable)     interior only (the boundary is non-continuous)
   A4:  H_edge(θ) ≥ H_min     entropy floor
   A5:  κ < κ_max(p, q)       κ cap from the holographic bound  }
```

v8.1 の観測「安定領域は p/q 中域、σ 小、κ 中程度」は、v10 で実際の相図として得られる（図 3）。p の上限は A2（飽和）が、q の下限は A4（孤立化）が与える。閾値（r_min = 0.15, r_max = 0.85, H_min = 0.94 log n）は v1〜v9 の経験から定めた暫定値であり、§10 で述べる通り、将来はホログラフィック境界エントロピーからの導出に置き換えられるべきものである。

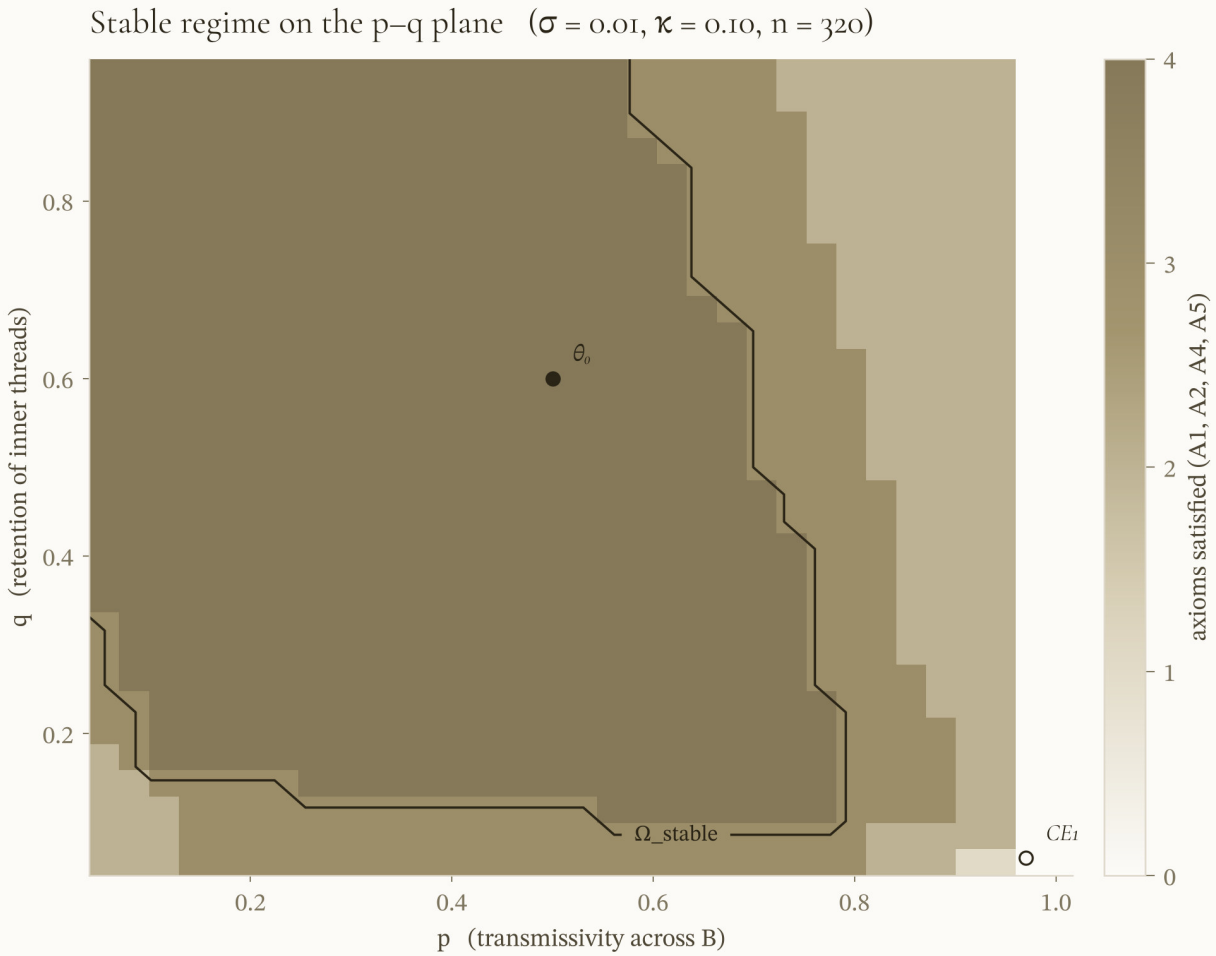

図 3　p–q 平面の安定領域（実計算・σ = 0.01、κ = 0.10、n = 320）。濃い領域が Ω_stable、θ_0 = (0.5, 0.6) は v10 plate の設定、右下の CE1 は飽和。

— 4.5　公理の検証関数

```
五公理の検査
def check_axioms(theta, obs):
    # obs は観測量の辞書：entropy, depth, cross, area, n
    p, q, s, k = theta.p, theta.q, theta.sigma, theta.kappa
    # A1 順序保存：揺らぎ σ が最小リンク長（≈ 1/√n）より小さいこと
    a1 = s < 0.5 / math.sqrt(obs["n"])
    # A2 地平面の二相性：越える糸が途絶（二つの世界に分断）も飽和（地平面の消失）もしていないこと
    a2 = 0.15 < obs["cross_ratio"] < 0.85
    # A4 エントロピー非崩壊：次数エントロピーが床 H_min を上回ること（log n の 94%）
    a4 = obs["entropy"] > 0.94 * math.log(obs["n"])
    # A5 ホログラフィック境界：越える糸の本数 ≤ 地平面の面積（要素数）
    a5 = obs["cross"] ⩽ obs["area"]
    # A3 局所連続性は θ の近傍で判定するため、別関数 is_interior に委ねる
    return {"A1": a1, "A2": a2, "A4": a4, "A5": a5}
```

```
def observe(pts, L, inside, theta, seed=0):
    # 投影規則 f を適用してホログラム H と糸 E を得る
    H, E = project(pts, L, inside, theta, seed)
    # 要素数
    n = len(pts)
    # 越える糸の本数（投影後）
    cross = int((inside[E[:, 0]] ≠ inside[E[:, 1]]).sum()) if len(E) else 0
    # 投影前の越える糸の本数（比率の分母）
    cross0 = max(cross_links(L, inside)[0], 1)
    # 観測量を辞書にまとめて返す
    return H, E, {"n": n, "entropy": edge_entropy(E, n), "depth": reachability_depth(E, n),
                  "cross": cross, "cross_ratio": cross / cross0, "area": horizon_area(L, inside)}
```

```
def is_interior(pts, L, inside, theta, eps=0.04):
    # A3：θ を eps だけ動かした 4 方向すべてで A1, A2, A4, A5 が保たれるなら内部点
    for dp, dq in [(eps, 0), (-eps, 0), (0, eps), (0, -eps)]:
        # 近傍の θ′ を作る（範囲外に出ないよう 0.01〜0.99 に留める）
        t2 = Controller(min(max(theta.p + dp, 0.01), 0.99), min(max(theta.q + dq, 0.01), 0.99),
theta.sigma, theta.kappa)
        # θ′ で観測し、公理を検査する
        ok = check_axioms(t2, observe(pts, L, inside, t2)[2])
        # 一つでも破れたら境界上にいる（相転移点）
        if not all(ok.values()):
            # 内部点ではない
            return False
    # 四方向とも保たれた：安定領域の内部
    return True
```

```
def is_stable(pts, L, inside, theta):
    # A1, A2, A4, A5 を θ 自身で検査する
    ok = check_axioms(theta, observe(pts, L, inside, theta)[2])
    # A3 を近傍で検査する
    ok["A3"] = is_interior(pts, L, inside, theta)
    # 五つすべてが真のときだけ、θ は Ω_stable に属する
    return all(ok.values()), ok
```

```
相図の計算
def phase_diagram(pts, L, inside, sigma=0.01, kappa=0.1, grid=25):
    # p と q を (0, 1) の格子で走査する
    ps = np.linspace(0.04, 0.96, grid)
    # q も同じ格子
    qs = np.linspace(0.04, 0.96, grid)
    # 各格子点で「満たした公理の数」を記録する
    S = np.zeros((grid, grid))
    # 各格子点でのエントロピーも記録する（図の等高線に使う）
    Hn = np.zeros((grid, grid))
    # 二重ループで走査
    for a, q in enumerate(qs):
        # 内側のループは p
        for b, p in enumerate(ps):
            # この格子点の θ
            th = Controller(p, q, sigma, kappa)
            # 観測して公理を検査する（A3 は高価なので相図では省く）
            _, _, obs = observe(pts, L, inside, th)
            # 真の公理の数 (0〜4)
            S[a, b] = sum(check_axioms(th, obs).values())
            # エントロピーを log n で割って 0〜1 に正規化
            Hn[a, b] = obs["entropy"] / math.log(len(pts))
    # 格子と二つの場を返す
    return ps, qs, S, Hn
```

— 4.6　f の一意性について

定理（非形式的）：安定領域 Ω_stable 内において、A1〜A5 を満たす写像 f は同型を除いて一意である。これは「葬列という構造は、これらの制約から一意に定まる」という主張である。物理的に言えば、ホログラフィック原理において境界データから bulk の構造が一意に復元される、ということである。仏教的に言えば、阿頼耶識（蔵識）から現象世界が一意に展開される。アート的に言えば、seed と θ が固定されれば plate は一意に決まる——それが「再現可能な制作工程」の意味である。

証明の方針：(1) f が A1〜A5 を満たすと仮定する。(2) 任意の S, B, $\theta \in \Omega$_stable に対して、f(S; B, θ) の構造的特徴量（reachability depth、edge entropy、g(r)）を計算する。(3) これらの特徴量が θ のみに依存し、S の選択に依存しないことを示す（普遍性）。(4) 二つの写像 f, f′ が同じ特徴量を持つなら、同型 ϕ が存在して f′ = $\phi \circ$ f が成り立つことを示す。v10 で (2) は計算可能になった。(3) と (4) は未証明のまま、§10 の課題として残す。

第 5 章

v10 —— 地平面の内側

— 5.1　ペンローズ図と事象の地平面

シュヴァルツシルト時空の最大拡張をペンローズ図に描くと、外部領域 I と地平面の内側 II が、45 度のヌル線 B で分かれる（図 4）。B は光の経路であり、B に沿って固有時は流れない。内側の世界線は、外から見れば B に漸近して止まり、内から見れば特異点 r = 0 に向かう。本稿が扱うのは、内側の世界線が f の定義域（Ω_stable）を出る前に記述を終える、そのような B である。これは静的なシュヴァルツシルト地平面の理想化であり、蒸発（ホーキング放射）と特異点は範囲外に置く。特異点は f が定義されない場所であり、本稿はそこについて何も主張しない。

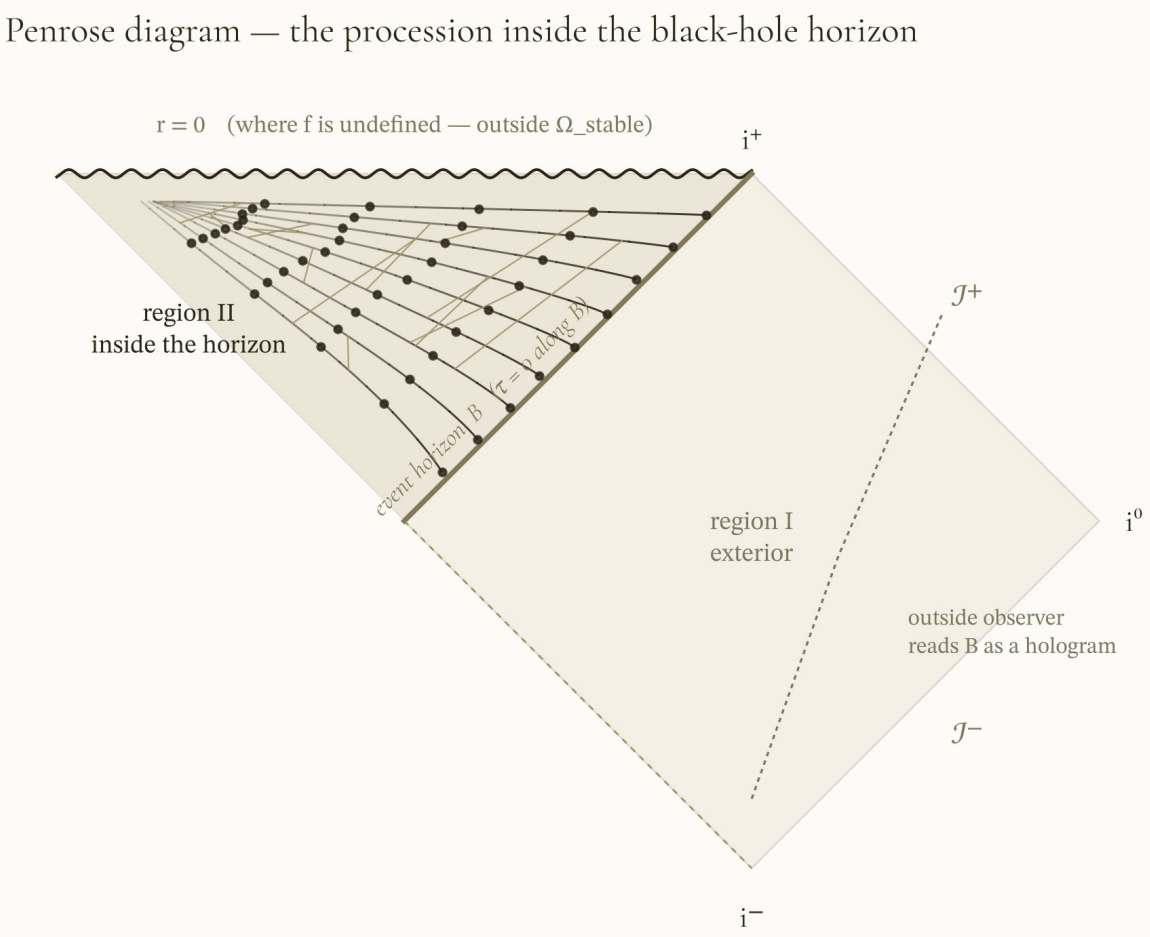

図 4　ペンローズ図——ブラックホールの地平面の内側にある葬列。外部の観測者は B をホログラムとして読む。

命題 P2 の内容はこうである。私たちが「いま」と呼ぶものは、地平面を過去に持つ領域の内側で、互いに糸で結ばれた要素の集合である。外側の観測者にとって、私たちの出来事はすべて B 上に投影された情報として読まれる。私たちにとって、B は越えられない過去である。両者は同じ構造の二つの読み方であり、どちらかが本当でどちらかが幻だということはない。

— 5.2　ビット・スレッドと κ_max の導出

Freedman と Headrick は、ホログラフィックなエントロピー公式（Ryu–Takayanagi）を「極小曲面」ではなく「最大流」として言い換えた。境界領域のエントロピーは、その領域から流れ出せる独立な糸（bit threads）の最大本数であり、最大流・最小カット定理によって、それは糸を切る最小の面——地平面——の面積に等しい。本稿の A5 は、この言い換えの離散版である。

```
BIT THREADS による K_MAX の導出
def max_threads_across(E, inside, n):
    # 無向グラフを作る（糸は向きを問わず「流せる」）
    G = nx.Graph()
    # 全要素を頂点に
    G.add_nodes_from(range(n))
    # 各糸に容量 1 を与える。1 本の糸は 1 ビットしか運べない
    for a, b in E:
        # 既にある辺の容量を読む（なければ 0）
        c0 = G.get_edge_data(int(a), int(b), {"capacity": 0})["capacity"]
        # 容量を 1 増やす（多重辺は束ねる）
        G.add_edge(int(a), int(b), capacity=c0 + 1)
    # 超頂点 S を内側のすべての要素に、T を外側のすべての要素に無限容量で結ぶ
    S, T = n, n + 1
    # 超頂点を先に登録しておく（外側が空でも T が存在するように）
    G.add_nodes_from([S, T])
    # 内側へ
    G.add_edges_from((S, int(v), {"capacity": float("inf")}) for v in np.where(inside)[0])
    # 外側へ
    G.add_edges_from((int(v), T, {"capacity": float("inf")}) for v in np.where(~inside)[0])
    # 最大流 = 内側から外側へ同時に流せるビット数 = 最小カット（地平面の「面積」）
    flow, _ = nx.maximum_flow(G, S, T)
    # 整数で返す
    return int(flow)
```

```
def kappa_max(pts, L, inside, p, q, sigma=0.01, trials=12):
    # κ を 0 から増やしながら、A5（越える糸 ≤ 面積）が破れる最小の κ を探す
    area = horizon_area(L, inside)
    # κ の候補
    for k in np.linspace(0.0, 1.5, trials):
        # この κ で観測
        _, E, obs = observe(pts, L, inside, Controller(p, q, sigma, k))
        # 最大流で「本当に通る糸」を数える（単に越える本数ではなく、独立に流せる本数）
        if max_threads_across(E, inside, len(pts)) > area:
            # 面積を超えた：ここが κ_max
            return float(k)
    # 走査範囲内では破れなかった
    return 1.5
```

κ_max(p, q) は「越える糸の最大流が地平面の面積を超える最小の κ」として数値的に定まる。

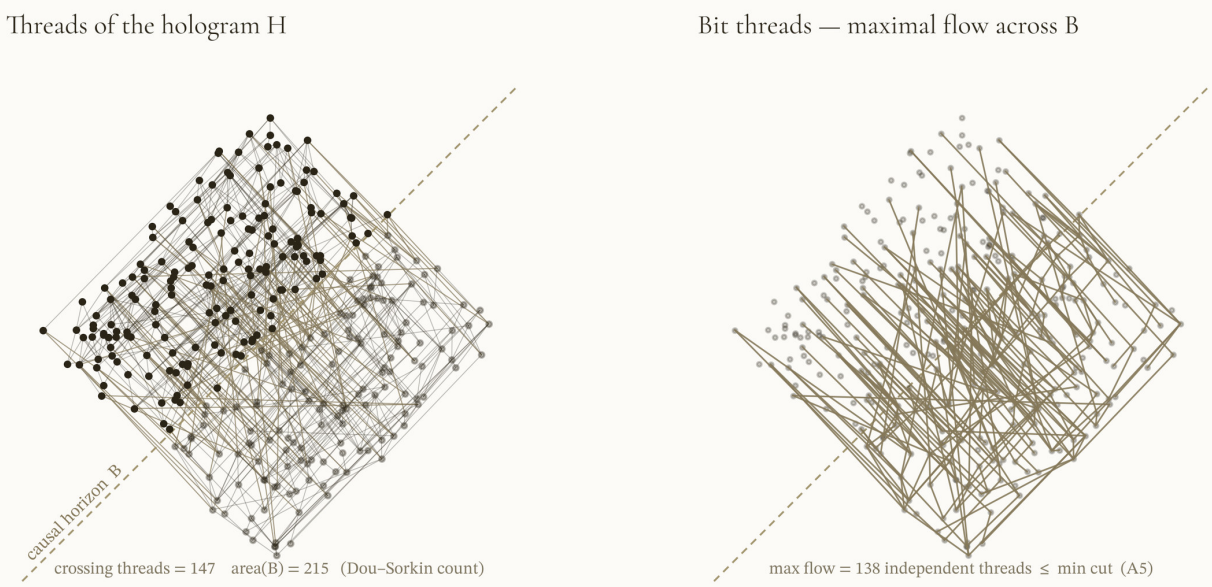

図 5 ホログラム H の糸（左）と、地平面を越える最大流としてのビット・スレッド（右）。最大流は最小カット＝面積を超えない。

v9 では κ_max を「p, q に依存する上限」とだけ書いた。v10 では $\theta_0 = (0.5, 0.6)$ で κ_max ≈ 0.27、p を下げて (0.3, 0.5) とすると κ_max ≈ 0.41 となる。透過率 p が低いほど、地平面の面積のうち干渉に使える余地が増える——これは Bekenstein 境界が「面積のうち既に使われている分」を差し引いた残りを許す、という読み方と一致する。

— 5.3 Dou–Sorkin の地平面エントロピー

Dou と Sorkin は、因果集合におけるブラックホールのエントロピーを「地平面を横切るリンクの本数」で数えることを提案した。本稿が地平面の面積をその本数で測るのはこの提案に従ったものであり、A5 は「越える糸は、面積として数えられたリンクを超えて増やせない」という自明に近い形になる。自明に近いことが重要である。公理は、できるだけ当たり前であるべきだからだ。

— 5.4 面積則の検証——素朴な数え方では破れる

A5 の物理的な根拠は面積則である。地平面を横切る糸の本数が、体積（要素数 n）ではなく面積に比例するなら、1+1 次元では長さ ∝ √n になるはずだ。要素数を 80 から 1280 まで変えて横切るリンクを数えると、両対数の傾きは 1.04——面積ではなく体積に比例する（図 12-II）。理由は 1+1 次元の因果集合に固有である：リンクはほとんどヌルで長く、地平面の近傍に集中しない。Dou と Sorkin が面積則を得たのは、空間的な超曲面 Σ を介して「Σ の手前で極大・Σ の先で極小」の対だけを数える精密化によるものであり、本稿の素朴な数え方はそれを欠く。この否定的な結果は、§10 に挙げた二つの課題——閾値の解析的導出と 3+1 次元化——が同じ一つの問題であることを示している。

```
面積則の検証
def area_law(ns=(80, 160, 320, 640, 1280), seed=7):
    # 要素数 n を変えながら、地平面を横切るリンクの本数を数える
    out = []
    # 各 n で
    for n in ns:
        # 散布・順序・リンク・地平面
        pts = sprinkle_diamond(n, seed); L = links(causal_matrix(pts)); inside = horizon_split(pts)
        # 横切るリンク数を記録
        out.append((n, cross_links(L, inside)[0]))
    # 1+1 次元では地平面の「面積」は長さ ∝ √n。横切るリンク数が n^0.5 で増えれば面積則、n^1 なら体積則
    xs = np.log([a for a, _ in out]); ys = np.log([b for _, b in out])
    # 両対数の傾き（最小二乗）
    slope = float(np.polyfit(xs, ys, 1)[0])
    # 測定値と傾きを返す。素朴な数え方では傾きは 1 に近い：1+1 次元のリンクはほぼヌルで長く、地平面の近傍に集中しないからである
    return out, slope
```

第 6 章

弦と膜 —— 超弦理論・M 理論からの読み替え

ここからの二章は、定理ではなく対応を述べる。対応とは「構造が同じ形をしている」という観察であり、同じ物であるという主張ではない。どこまでが計算され、どこからが読み替えかを、各節の末尾に明記する。

— 6.1　糸は弦である

本稿の「因果の糸」は、二つの要素を結ぶ一次元の対象である。弦理論の基本対象もまた一次元の弦であり、その世界面は時空の中に二次元の面を描く。ビット・スレッドが「切れない一本の糸」として境界を貫くとき、それは弦の世界面の断面にほかならない。§5.2 の最大流は、地平面を貫く弦の最大本数と読める。

— 6.2　T 双対と「大きさのない世界」

閉弦を半径 R の円に巻くと、質量スペクトルは運動量の量子数 n と巻き付き数 w で決まり、R を α'/R に置き換えても n と w を入れ替えるだけで同じスペクトルが得られる。これが T 双対である。「大きい円」と「小さい円」は区別できない。本稿の文脈では、これは命題 P1 の弦理論版である：窓（R）の大きさは物理ではない。始まりと終わりが窓の縁の産物であるように、大小は巻き付きの読み替えにすぎない。

```
弦の双対性：T 双対と質量スペクトル
def closed_string_spectrum(R, alpha=1.0, nmax=3, wmax=3):
    # 半径 R の円に巻いた閉弦の質量² = (n/R)² + (wR/α′)² + 振動（ここでは省略）
    spec = set()
    # 運動量量子数 n（円周方向の波の数）
    for n in range(-nmax, nmax + 1):
        # 巻き付き数 w（弦が円に何回巻いているか）
        for w in range(-wmax, wmax + 1):
            # 質量² を小数 6 桁に丸めて集合に入れる
            spec.add(round((n / R) ** 2 + (w * R / alpha) ** 2, 6))
    # 質量² の集合を昇順の組で返す
    return tuple(sorted(spec))
```

```
def t_duality_holds(R, alpha=1.0):
    # T 双対：半径 R の弦と半径 α′/R の弦は同じスペクトルを持つ（n と w が入れ替わる）
    return closed_string_spectrum(R, alpha) == closed_string_spectrum(alpha / R, alpha)
```

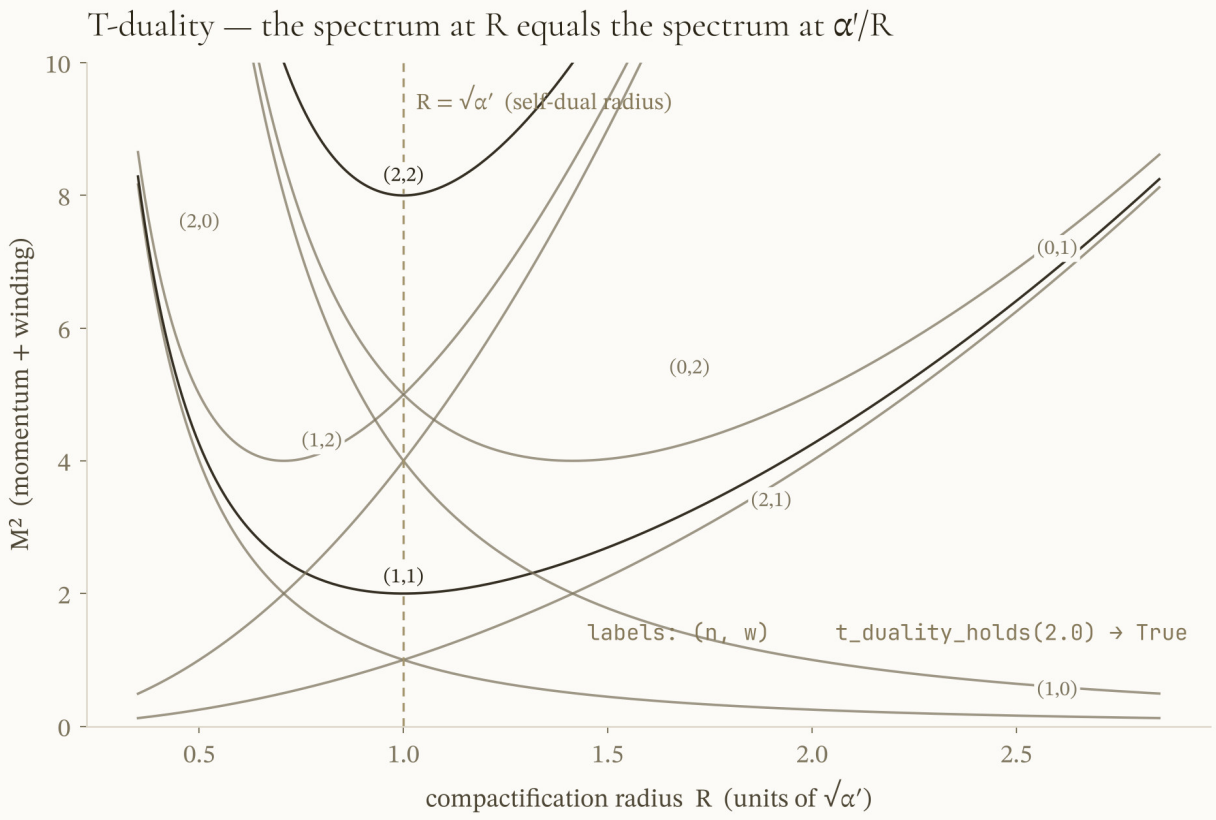

図 7　T 双対——半径 R のスペクトルは α′/R のスペクトルに等しい。自己双対半径 R = √α′ を軸に、(n, w) と (w, n) が鏡像になる。

— 6.3　M 理論とブレーン上の世界

五つの超弦理論は、11 次元の M 理論の異なる極限として統合された。そこでは弦だけでなく膜（ブレーン）が基本対象になり、開弦の端はブレーンに固定される。ブレーン宇宙論は「私たちの 4 次元世界は高次元時空に浮かぶブレーンである」と考え、AdS/CFT 対応は「境界上の理論が bulk の重力を完全に記述する」ことを示した。本稿の地平面 B は、この意味でのブレーンである。私たちはブレーン（B）の上に住んでいるのではなく、B を過去に持つ内側にいて、B 上に投影される。6.1〜6.3 で計算されたのは T 双対の式だけであり、残りは対応である。

本稿	弦理論・M 理論	身分
因果の糸	弦の世界面の断面	対応
ビット・スレッドの最大流	地平面を貫く弦の本数	対応
地平面 B	ブレーン／AdS 境界	対応
窓の大きさは物理でない（P1）	T 双対 R ↔ α′/R	計算（§6.2）
A5 面積則	Bekenstein–Hawking・Ryu–Takayanagi	離散アナログ

第 7 章

曼荼羅と生命の樹 —— 順序の同心投影

7.1 生命の樹は部分順序である

カバラの生命の樹は、十のセフィラと二十二の径から成る。径に「流出」の向きを与えると（ケテルからマルクトへ）、樹は有向非巡回グラフ、すなわち部分順序になる。各セフィラの rank——ケテルからの最長の径の長さ——を計算すると 0 から 8 まで九段になり（図 6 左）、推移閉包を除いた「リンク」だけが太く残る。v2 以来「Sephirot 制約」と呼んできたものは、この順序を因果集合の上に重ねる操作であった。v10 でようやく、制約の中身が「十要素の部分順序との準同型」として書ける。

```
生命の樹（セフィロト）の順序構造
SEPHIROTH = ["Kether", "Chokmah", "Binah", "Chesed", "Geburah", "Tiphereth", "Netzach", "Hod",
"Yesod", "Malkuth"]
# 22 の径（パス）。各組 (a, b) は「a から b へ流れ出す」向きで記す（流出の順序）
PATHS = [(0, 1), (0, 2), (0, 5), (1, 2), (1, 3), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5), (3, 6),
         (4, 5), (4, 7), (5, 6), (5, 7), (5, 8), (6, 7), (6, 8), (6, 9), (7, 8), (7, 9), (8, 9)]
```

```
def tree_of_life():
    # 有向グラフとして生命の樹を作る
    G = nx.DiGraph(PATHS)
    # 非巡回なら「順序」になる（流出に逆流がない）
    acyclic = nx.is_directed_acyclic_graph(G)
    # 各セフィラの rank = Kether からの最長径の長さ
    rank = {v: nx.dag_longest_path_length(G.subgraph(nx.ancestors(G, v) | {v})) for v in G}
    # 推移閉包から「リンク」（中継のない径）だけを残す
    TC = nx.transitive_closure_dag(G)
    # 径 (a, b) がリンク ⇔ a→c→b となる c が存在しない
    link = [(a, b) for a, b in PATHS if not any((a, c) in TC.edges and (c, b) in TC.edges for c in
G if c not in (a, b))]
    # 結果をまとめて返す
    return acyclic, rank, link
```

7.2 曼荼羅投影

曼荼羅は中心から外へ向かう同心の構造である。因果集合の rank を半径に、空間座標 x を円周に巻き付けて（コンパクト化——§6.2 の円と同じ操作である）投影すると、過去を持たない要素が中心に集まり、最も深い段が外周に並ぶ（図 6 右）。因果の糸は放射状の線になる。胎蔵曼荼羅の中央に大日如来が座し、そこから諸尊が外へ展開する構図と、この投影は同じ形をしている。中心は「原因のない原因」ではなく、窓の中で過去を持たない要素——つまり窓の縁（§8.1）である。曼荼羅の中心は、観測の限界が描く点である。

```
曼荼羅投影：RANK を同心円へ
def mandala_layout(pts, rank):
    # 最大 rank。これが外周の環になる
    rmax = max(int(rank.max()), 1)
    # 半径 = rank / rmax。rank 0（過去を持たない要素）が中心に、最も深い段が外周に来る
    radius = rank / rmax
    # 空間座標 x を円周に巻き付ける：x の範囲 [−a, a] を角度 [0, 2π) に写す（コンパクト化）
    a = np.abs(pts[:, 1]).max() + 1e-9
    # 角度 θ = π (x + a) / a
    theta = math.pi * (pts[:, 1] + a) / a
    # 極座標を直交座標へ。因果の糸は中心から外へ向かう放射状の線になる
    return np.column_stack([radius * np.cos(theta), radius * np.sin(theta)])
```

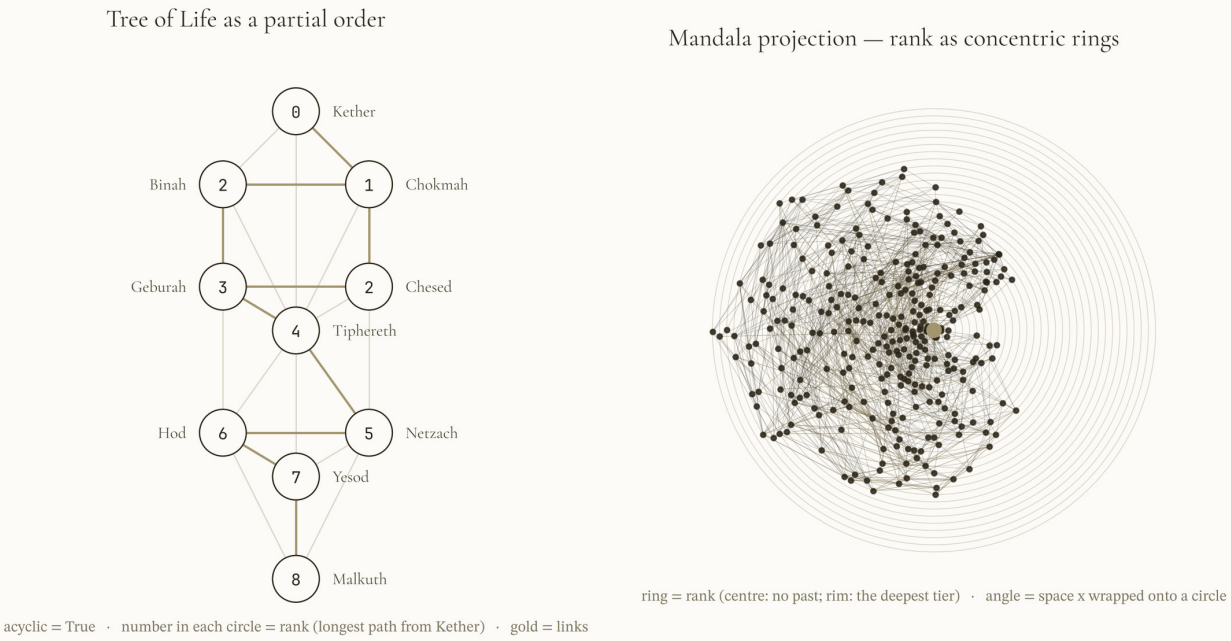

図 6　左——部分順序としての生命の樹（丸の中の数字が rank、金の線がリンク）。右——曼荼羅投影（環が rank、角度が円に巻いた空間 x）。

7.3　両界と二相性

真言密教は胎蔵界と金剛界の二つの曼荼羅を対にして掲げる。本稿の A2——地平面の内外が構造的に異なる二相を保つこと——は、一つの構造に二つの読み方があり、どちらも欠かせない、という形式を持つ。§5.1 で述べた「外側の観測者は B をホログラムとして読み、内側の私たちは B を越えられない過去として持つ」という二つの読みも、同じ形式である。これは対応であって定理ではない。

第 8 章

不生不滅 —— 涅槃の順序論的な言い直し

— 8.1　始まりと終わりは窓の縁である

有限の因果集合には必ず極小元（過去を持たない要素）と極大元（未来を持たない要素）がある。それを「始まり」「終わり」と呼ぶのは自然である。だが、観測窓を固定したまま、散布する領域だけを広げると、窓の中の要素のうち過去を持たないものの割合は急速に 0 に向かう（図 8）。領域を 1.2 倍にしただけで、「生まれた」要素はいなくなる。始まりと終わりは順序の性質ではなく、順序を切り取った窓の性質である。

```
不生不滅の判定
def unborn_unending(C):
    # 各要素に過去があるか（列ベクトルに真が一つでもあるか）
    has_past = C.any(axis=0)
    # 各要素に未来があるか（行ベクトルに真が一つでもあるか）
    has_future = C.any(axis=1)
    # 「生まれていない」= 過去がある（最初の要素ではない）。「滅していない」= 未来がある
    return has_past, has_future
```

```
def null_proper_time(pts_on_horizon):
    # 地平面上の 2 点間の固有時 τ² = Δt² − Δx²。ヌル直線上では常に 0
    d = np.diff(pts_on_horizon, axis=0)
    # 各区間の τ²（浮動小数の誤差で ±1e−12 程度になる）
    tau2 = d[:, 0] ** 2 - d[:, 1] ** 2
    # 全区間の合計。「地平面には時間が流れない」ことの数値表現
    return float(tau2.sum())
```

```
def window_test(scales=(1.0, 1.02, 1.05, 1.1, 1.2, 1.35, 1.5, 2.0), density=320, seed=11):
    # 観測窓（単位ダイヤモンド）を固定し、散布する領域だけを scale 倍に広げていく
    out = []
    # 各スケールで試す
    for s in scales:
        # 密度を一定に保つため、要素数は面積（scale²）に比例させる
        pts = sprinkle_diamond(int(density * s * s), seed) * s
        # 広げた領域の中心が窓の中心に来るよう平行移動する
        pts[:, 0] -= (s - 1.0) * math.sqrt(2.0) / 2.0
        # 因果行列（広げた領域全体）
        C = causal_matrix(pts)
        # 窓の内側にある要素（元の単位ダイヤモンド：0 ≤ u, v < 1）
        u = (pts[:, 0] - pts[:, 1]) / math.sqrt(2.0)
        # v も同様
        v = (pts[:, 0] + pts[:, 1]) / math.sqrt(2.0)
        # 窓の判定
        win = (u >= 0) & (u < 1) & (v >= 0) & (v < 1)
        # 窓の要素のうち、広げた領域全体の中で過去を持たないものの割合
        no_past = float((~C[:, win].any(axis=0)).mean())
        # 同じく未来を持たないものの割合
        no_future = float((~C[win, :].any(axis=1)).mean())
        # (scale, 始まりの割合, 終わりの割合) を記録する
        out.append((s, no_past, no_future))
    # scale が大きくなるほど両者が 0 に近づく：始まりと終わりは窓の縁の産物である
    return out
```

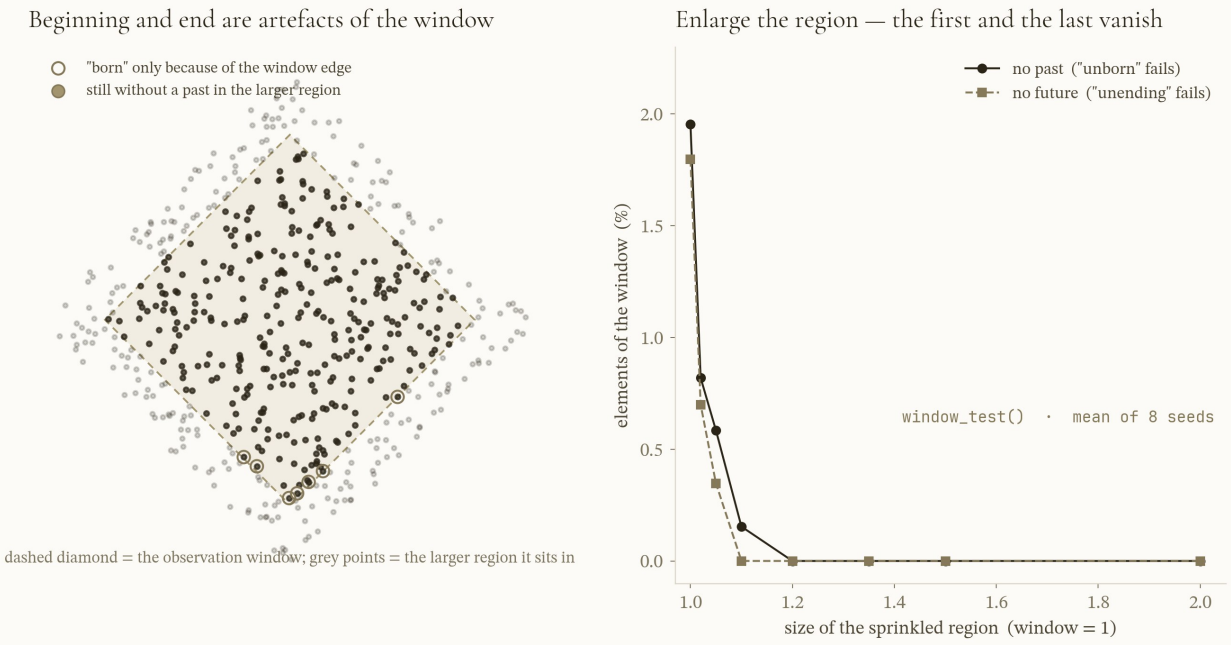

図 8　左――窓（破線のダイヤモンド）の縁で「生まれた」ように見える要素は、より広い領域では過去を持つ。右――領域を広げると、過去を持たない要素・未来を持たない要素の割合は 0 に向かう（8 seed の平均）。

— 8.2　地平面に時間は流れない

地平面はヌル面であり、その上の二点間の固有時は $\tau^2 = \Delta t^2 - \Delta x^2 = 0$ である。地平面に沿う視点にとって、内側のすべての出来事は同時である。作品の言葉——「過去・現在・未来の出来事が一瞬のうちに展開される」——は、地平面上の視点の定義そのものである。涅槃を「時間の外」と呼ぶ代わりに、本稿は「$\tau = 0$ の面上の視点」と呼ぶ。

— 8.3　縁起・無我・不生不滅

因果集合の要素は、順序以外の属性を持たない。要素が「何であるか」は、何の後にあり何の前にあるかで尽きる。これは無我（要素に固有の実体はない）の形式的な言い直しであり、縁起（すべては条件によって生じる）は順序 $\prec$ そのものである。般若心経の「不生不滅」は、§8.1 の窓の縁の定理に対応する。ブッダが至った涅槃を本稿は定義しない。定義できるのは、始まりと終わりが窓の産物であること、地平面上では時間が流れないこと、そしてすべての要素が列に属すること（§8.4）までである。その先は、計算ではなく修行の領域である。

— 8.4　すべての人々は葬列を成す

Dilworth の定理：有限な部分順序は、反鎖の最大の大きさ（幅）に等しい本数の鎖に分割できる。Fulkerson による証明は、二部グラフの最大マッチングを使う——「i の後に j が続ける」という辺を張り、できるだけ多くの要素に後続を割り当てると、後続を持たない要素の数が鎖の本数になる。320 要素の散布では 33 本の列が全要素を覆う（図 10）。どの要素もちょうど一つの列に属し、列の外に立つ者はいない。命題 P3 は、この定理の読み替えである。

```
葬列への分解（DILWORTH の定理）
def minimum_chain_cover(C):
    # 要素数
    n = C.shape[0]
    # 二部グラフを作る：左側に各要素の「出口」、右側に「入口」を置く
    G = nx.Graph()
    # 左右の頂点を登録する（左は 0..n−1、右は n..2n−1）
    G.add_nodes_from(range(2 * n))
    # i ≺ j なら、i の出口と j の入口を結ぶ（i の次に j が続けるという意味）
    i, j = np.where(C)
    # 辺を一括で登録する
    G.add_edges_from(zip(i, j + n))
    # 最大マッチング = 「次に誰かが続く」要素の最大数
    M = nx.bipartite.maximum_matching(G, top_nodes=range(n))
    # マッチした辺の数（辞書は両向きに入るので半分にする）
    matched = sum(1 for a in M if a < n)
    # Dilworth／Fulkerson：最小の鎖被覆数 = n − 最大マッチング。これが「葬列の本数」
    return n - matched, M
```

```python
def chains_from_matching(C, M):
    # 要素数
    n = C.shape[0]
    # 「誰かの後に続く」要素（右側でマッチした要素）を記録する
    follows = {M[b] : b - n for b in M if b ⩾ n}
    # 誰の後にも続かない要素が、各葬列の先頭である
    heads = [v for v in range(n) if v not in follows.values()]
    # 各先頭から、マッチングに沿って次の要素をたどり、鎖を作る
    chains = []
    # 先頭ごとに
    for h in heads:
        # 鎖は先頭から始まる
        chain = [h]
        # 出口がマッチしている限り次へ進む
        while chain[-1] in M and M[chain[-1]] ⩾ n:
            # 次の要素（右側の添字を元に戻す）
            chain.append(M[chain[-1]] - n)
        # 完成した鎖を追加する
        chains.append(chain)
    # すべての要素はちょうど一つの鎖に属する：誰も列の外にいない
    return chains
```

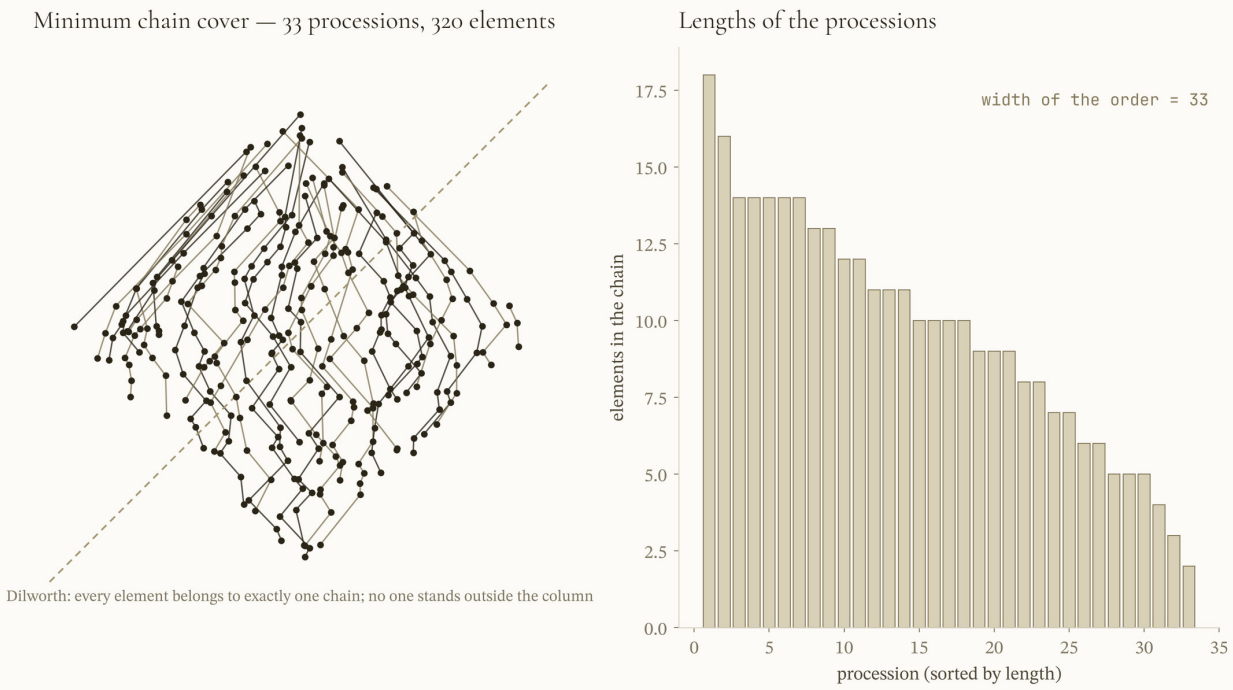

図 10　最小鎖被覆——320 要素が 33 本の葬列に分かれる（左）。各列の長さ（右）。

— 8.5　「今」は反鎖である

葬列の本数には双対がある。Dilworth の定理は「最小の鎖被覆数 = 最大の反鎖の大きさ」と述べる。反鎖とは、互いに因果関係を持たない要素の集合——どの二つも互いの過去にも未来にもない、つまり「同時」に在る要素の集合である。相対論に絶対的な同時刻面はないが、因果集合には「互いに無関係な要素の極大集

合」が常にあり、それが「今」の順序論的な定義になる。320 要素の散布では最大反鎖は 33 要素——葬列の本数とちょうど等しい（図 12-III）。同時に在ることのできる存在の数は、列の本数に等しい。一つの列に、一つの「今」。

```
「今」は反鎖である
def maximum_antichain(C, M):
    # 要素数
    n = C.shape[0]
    # 鎖被覆と同じ二部グラフを作り直す（左 = 出口、右 = 入口）
    G = nx.Graph(); G.add_nodes_from(range(2 * n)); i, j = np.where(C); G.add_edges_from(zip(i, j +
n))
    # König の定理：最大マッチングから最小頂点被覆が得られる
    cover = nx.bipartite.to_vertex_cover(G, M, top_nodes=range(n))
    # 出口も入口も被覆に入らない要素が、最大反鎖を成す（Dilworth の双対）
    anti = [v for v in range(n) if v not in cover and (v + n) not in cover]
    # 反鎖＝互いに因果関係のない要素の集合。どの二つも「同時」であり、その大きさ = 葬列の本数
    return len(anti), anti
```

— 8.6　縁——共通の未来

作品では、三つの像が一輪のバラを介して結ばれている。順序論で言えば、バラは共通の未来——二つの要素 i, j の両方の未来にある要素 k——である。「ご縁」を「i と j に共通の未来 k が存在すること」と定義すると、それは行列の積 $C\,C^{\mathsf{T}}$ で一度に計算できる。窓の中だけでは 93.9% の対がご縁を持ち、窓を固定して領域を 1.2 倍に広げると 99.9%、1.5 倍で 100% に達する（図 12-IV）。縁は、十分に広い世界では切れない。どの二人にも、必ず共通の未来がある。

```
縁＝共通の未来
def common_future_fraction(C):
    # 要素数
    n = C.shape[0]
    # F[i, j] = (i と j に共通の未来がある) = (ある k で i ≺ k かつ j ≺ k)
    Ci = C.astype(int)
    # 行列の積で一度に求める：(C Cᵀ)[i, j] = i と j の共通の未来の個数
    shared = (Ci @ Ci.T) > 0
    # 上三角（i < j）の対だけ数える
    iu = np.triu_indices(n, k=1)
    # 共通の未来を持つ対の割合
    return float(shared[iu].mean())
```

```
def en_window_test(scales=(1.0, 1.2, 1.5, 2.0, 3.0), density=320, seed=11):
    # 窓を固定して領域を広げ、窓の中の対が（領域全体の中で）共通の未来を持つ割合を見る
    out = []
    # 各スケール
    for s in scales:
        # 広げた領域に散布し、窓の中心に合わせる
        pts = sprinkle_diamond(int(density * s * s), seed) * s; pts[:, 0] -= (s - 1.0) *
math.sqrt(2.0) / 2.0
        # 因果行列
        C = causal_matrix(pts)
        # 窓の判定
        u = (pts[:, 0] - pts[:, 1]) / math.sqrt(2.0); v = (pts[:, 0] + pts[:, 1]) / math.sqrt(2.0)
        # 窓の中
        win = np.where((u ⩾ 0) & (u < 1) & (v ⩾ 0) & (v < 1))[0]
        # 窓の要素 i, j について、領域全体に共通の未来 k があるか
        Ci = C[win].astype(int)
        # 共通の未来の有無
        shared = (Ci @ Ci.T) > 0
        # 上三角の割合
        iu = np.triu_indices(len(win), k=1)
        # 記録
        out.append((s, float(shared[iu].mean())))
    # 領域が広がるほど、どの二人にも共通の未来がある——縁は切れない
    return out
```

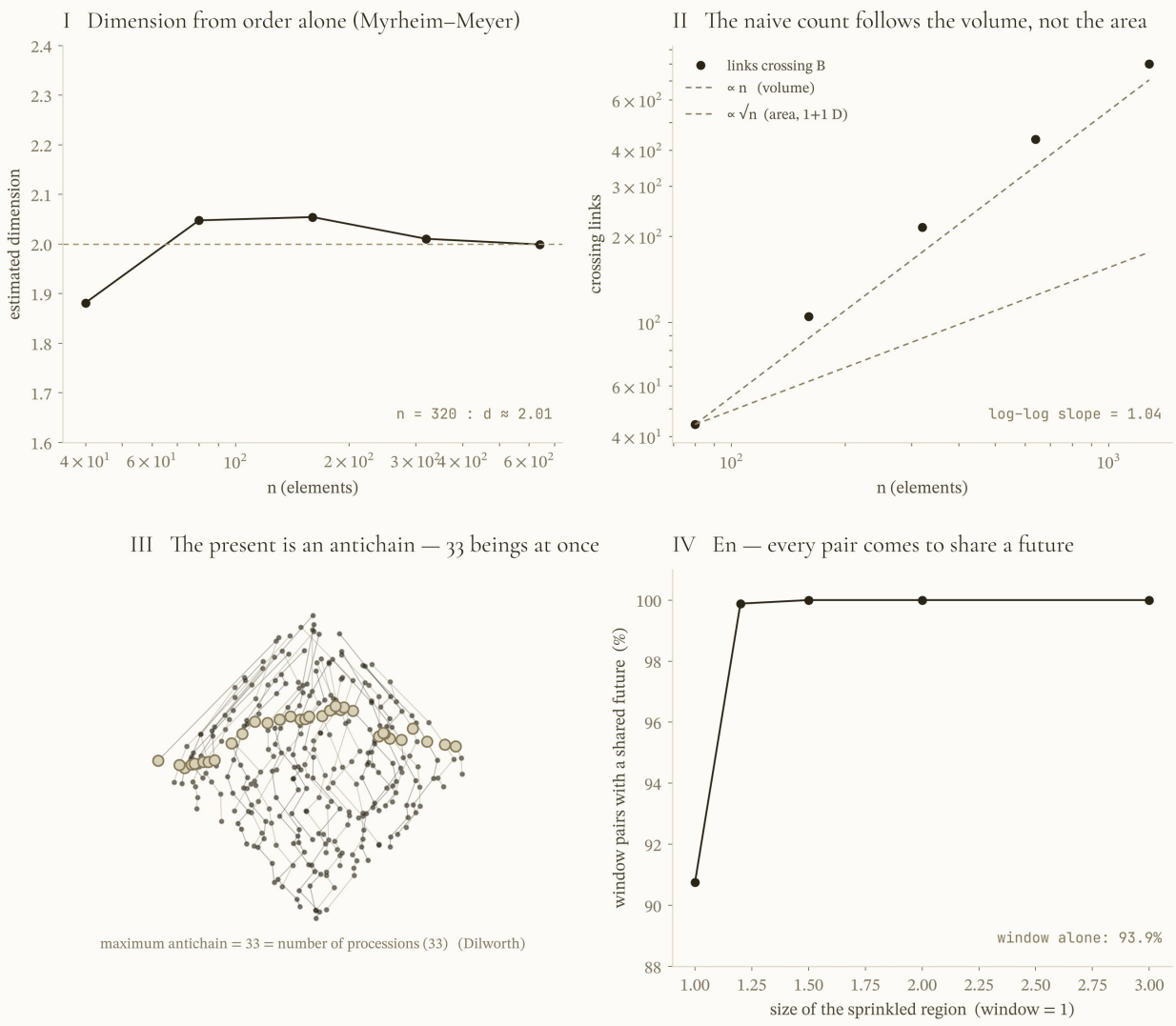

図 12 四つの拡張。I 順序だけから推定した次元（Myrheim–Meyer）。II 横切るリンク数は体積に比例する（面積則の否定的検証）。III 最大反鎖＝「今」（金の点・33 要素）。IV 窓の対のうち共通の未来を持つ割合。

第 9 章

文化進化と多様性 —— A4 の時間発展

A4 は「多様性が崩壊しないこと」を一瞬の構造について述べた。同じことを時間発展について述べると、文化進化論の中心問題になる。モラン過程——集団から一個体が複製され一個体が消える——は、変異がなければ必ず均質化し、型のエントロピーは床を割る。変異率 μ が小さくても正であれば、エントロピーは床の上で揺らぎ続ける（図 9）。葬列が「列」であり続けるためには、新しい糸が足され続けなければならない。A4 は保存則ではなく、供給の条件である。

```
文化進化：多様性の保全（A4 の時間発展）
def moran_entropy(n_types=8, pop=200, steps=400, mutation=0.02, seed=5):
    # 乱数
    rng = np.random.default_rng(seed)
    # 集団を n_types 種の型に等分して初期化する
    pop_state = np.repeat(np.arange(n_types), pop // n_types)
    # 各ステップのエントロピーを記録する
    H = []
    # モラン過程：1 個体が複製され、1 個体が消える
    for _ in range(steps):
        # 複製される個体（無作為）
        parent = rng.integers(len(pop_state))
        # 消える個体（無作為）
        gone = rng.integers(len(pop_state))
        # 確率 mutation で新しい型が生まれる（変異）。そうでなければ親の型を継ぐ
        pop_state[gone] = rng.integers(n_types) if rng.random() < mutation else pop_state[parent]
        # 型の頻度分布
        P = np.bincount(pop_state, minlength=n_types) / len(pop_state)
        # シャノンエントロピーを記録（0 に向かえば「均質化」= A4 の違反）
        H.append(float(-(P[P > 0] * np.log(P[P > 0])).sum()))
    # 時系列を返す
    return np.array(H)
```

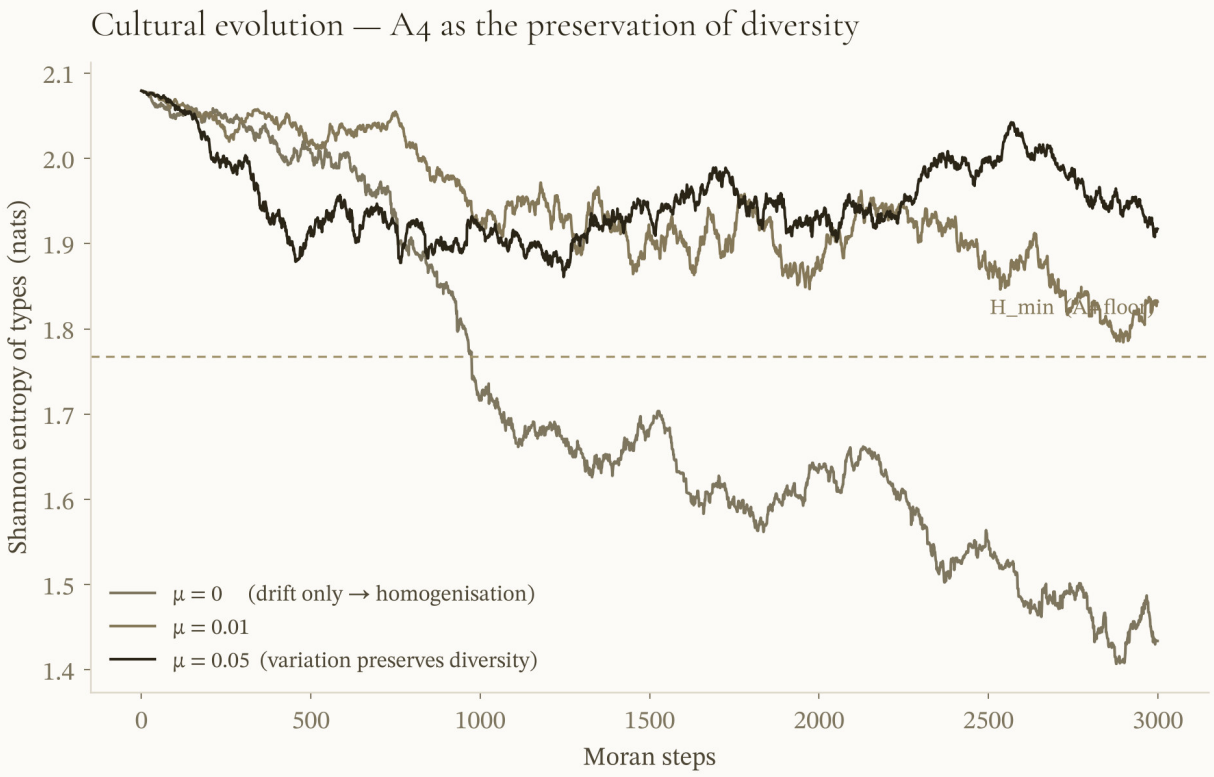

図 9 文化進化——モラン過程における型のエントロピー。変異がなければ均質化し（A4 違反）、小さな変異が多様性を保つ。

第 10 章

検証可能性と限界

本稿で計算されたこと：散布された因果集合の順序・rank・リンク、ヌル面による二分と越える糸の一方向性、f の実装、観測量、A1・A2・A4・A5 の検査と A3 の近傍検査、相図、最大流による κ_max、T 双対の式、生命の樹の順序構造、曼荼羅投影、窓の縁の定理の数値確認、$\tau = 0$、Dilworth の鎖分解、最大反鎖、共通の未来の割合、Myrheim–Meyer 次元、横切るリンクの n 依存性、モラン過程のエントロピー。これらはすべて付属のコードで再現できる。

本稿で計算されなかったこと：(i) 閾値 r_min, r_max, H_min, σ_max の導出。現在は経験値であり、Dou–Sorkin の数え方から解析的に出すべきである。(ii) f の一意性（§4.6 の (3)(4)）。(iii) 1+1 次元から 3+1 次元への拡張。地平面の「面積」が本当に面積になるのは 3+1 次元からであり、§5.4 の通り 1+1 次元では素朴な数え方が面積則を再現しない。(iv) 弦・ブレーンとの対応の定式化。(v) 涅槃そのもの。本稿は、計算できるところまでを計算し、できないところを「できない」と書くことで、作品と仕様の乖離を最小にした。真実に近づく唯一の方法は、どこまでが分かっていてどこからが分かっていないかを正確に言うことである。

第 11 章

結語 —— v10 が閉じるもの、開くもの

v9 は「葬列とは、五つの公理を満たす唯一の構造として時空に刻まれた因果の連なりである」と書いた。v10 はその文を、計算できる形で言い直した。葬列とは、ヌル面を過去に持つ領域の中で、Dilworth の鎖分解によって全要素を覆う列の束であり、その列を越える糸の本数は地平面の面積を超えず、列の最初と最後は観測窓の縁にしか現れない。詩と数学が同じことを言っている——v1 から追いかけてきたこの直観は、v10 で三つの定理的事実（窓の縁・$\tau = 0$・鎖分解）に着地した。

開いたままのもの：閾値の導出、一意性の証明、3+1 次元、弦との対応の定式化。そして一つ、計算の外にあるもの——列に属する一人ひとりが、互いの糸を大切にするかどうかである。それは公理ではなく、作品が問いかけ続けることだ。

葬列とは、地平面を過去に持つ世界の中で、誰も列の外に立たない順序のことである。

第 12 章

plate ギャラリー（v1〜v10）

各 plate には、(i) パラメータ（p, q, σ, κ）、(ii) 観測変化、(iii) 意図 を付与する。v1〜v9 の plate は、本稿の描画系（v10 の投影規則）で当時のパラメータから描き直し、全図の様式・書体・配色を揃えた。原図は `Past/` に保存している。v8〜v9 の図式 plate も同じ意匠で再構成した。



図 11 v10 plate——I 散布された因果集合と地平面、II ホログラム H = f(S; B, θ_0)、III 地平面で分けた二点相関、IV rank の分布。θ_0 = (0.5, 0.6, 0.01, 0.10)、n = 320、seed = 7。

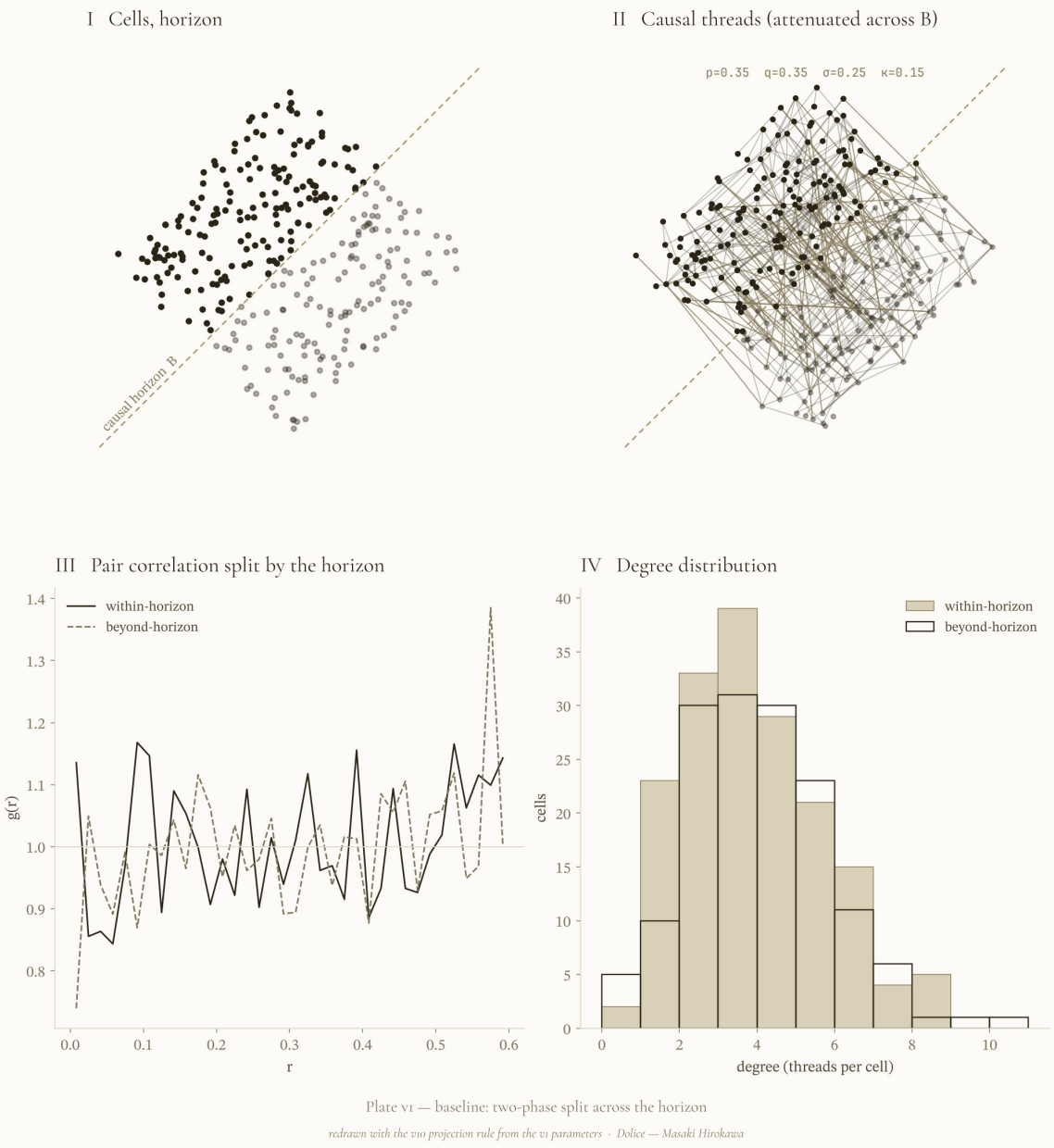

Plate v1 — baseline: two-phase split across the horizon

redrawn with the v10 projection rule from the v1 parameters · Dolice — Masaki Hirokawa

図 13 Plate v1 (baseline) — two-phase split across the horizon

パラメータ	`p = 0.350  q = 0.350  σ = 0.250  κ = 0.150`
観測	地平上下の二相が明確（structured / random）
意図	比較基準となる最小骨格を固定する

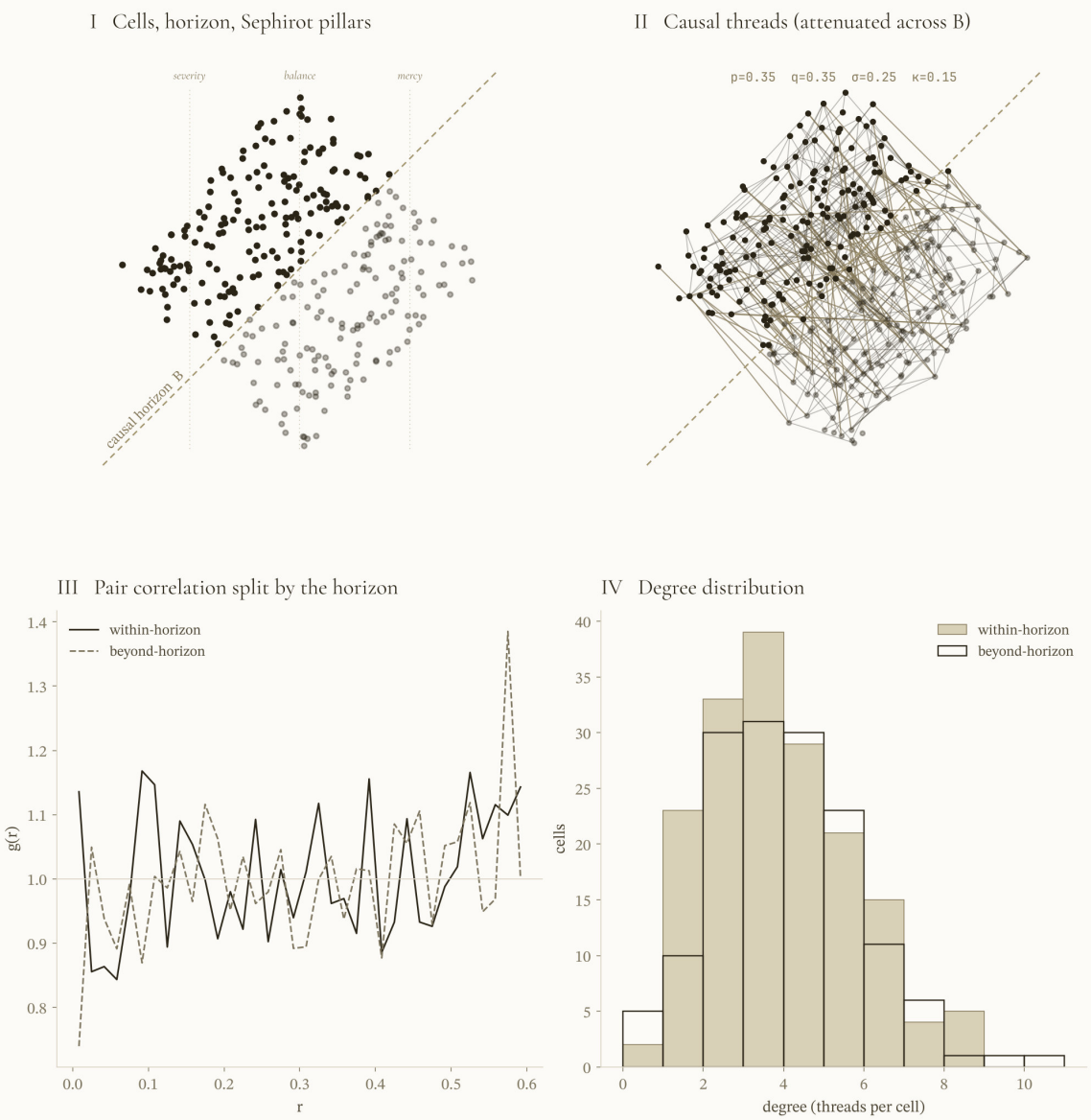

Plate v2 — Sephirot constraints: coherent density, increased order

redrawn with the v10 projection rule from the v2 parameters · Dolice — Masaki Hirokawa

図 14 Plate v2 — coherent density and increased order under constraints

パラメータ	`p = 0.350  q = 0.350  σ = 0.250  κ = 0.150`
観測	制約により密度が整い、秩序が増す
意図	Sephirot 制約で視覚上の「因果」を保つ

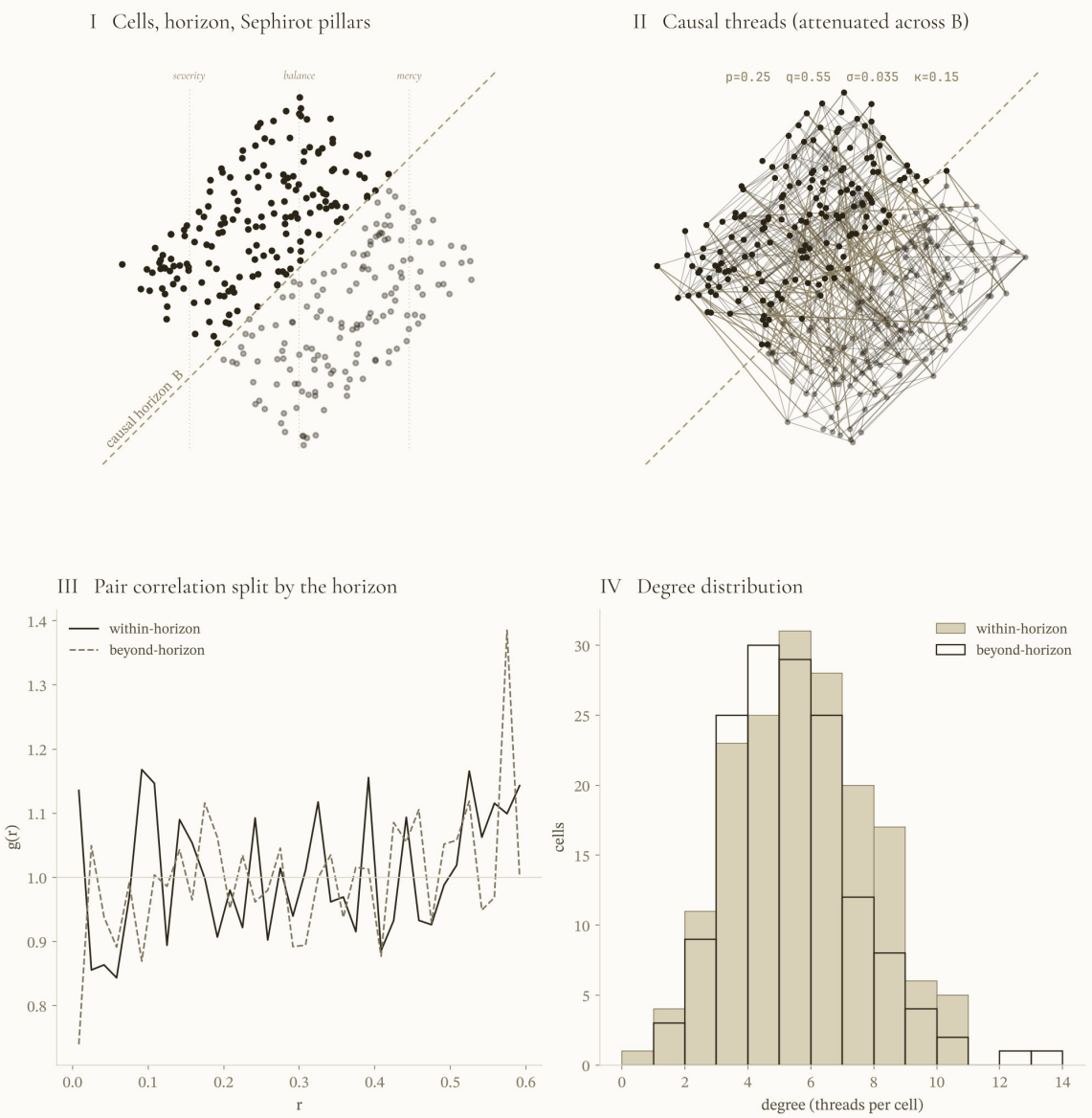

図 15 Plate v3 — locality and haze in balance (p–q shift)

パラメータ	`p = 0.250  q = 0.550  σ = 0.035  κ = 0.150`
観測	局所性と霧化のバランスが出る
意図	p, q, σ で硬さと曖昧さを同居させる

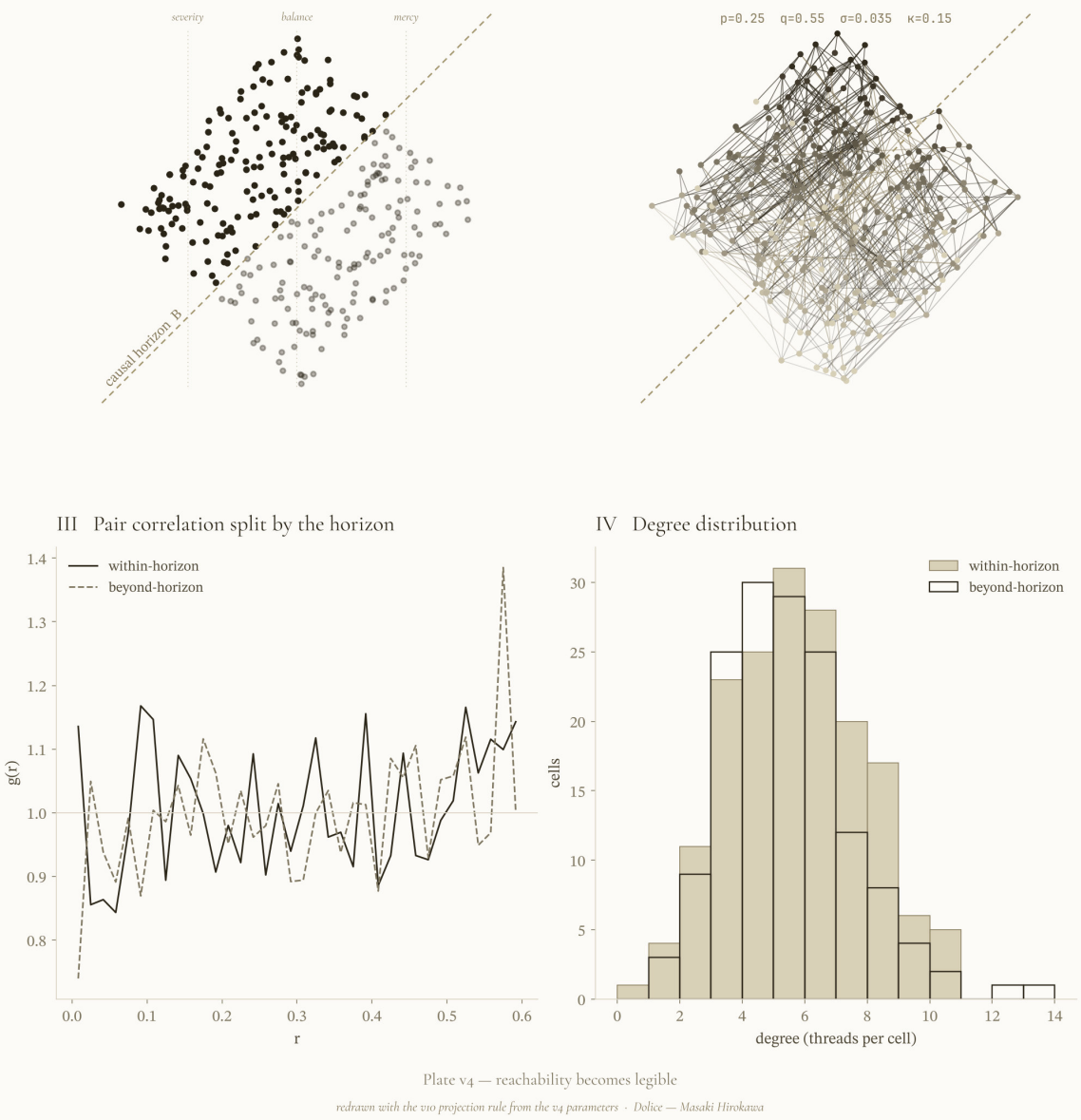

図 16 Plate v4 — reachability becomes more legible

パラメータ	`p = 0.250  q = 0.550  σ = 0.035  κ = 0.150`
観測	到達可能性の読みが明確になる
意図	因果の方向性を可視化へ反映する

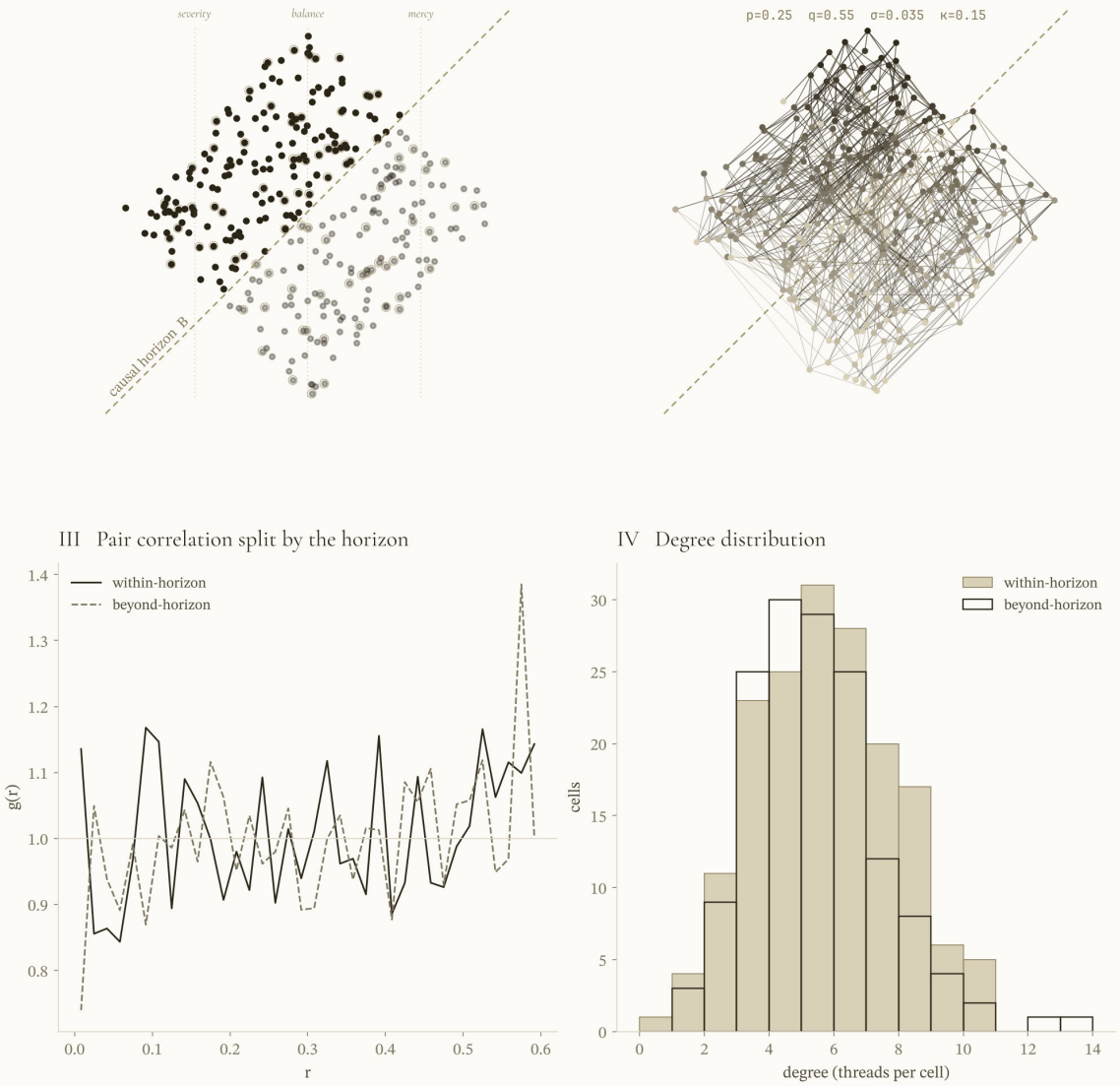

図 17 Plate v5 — rank repetition becomes prominent

パラメータ	`p = 0.250  q = 0.550  σ = 0.035  κ = 0.150`
観測	rank の反復が強く現れる
意図	葬列の「列」を入れ子構造として組織化する

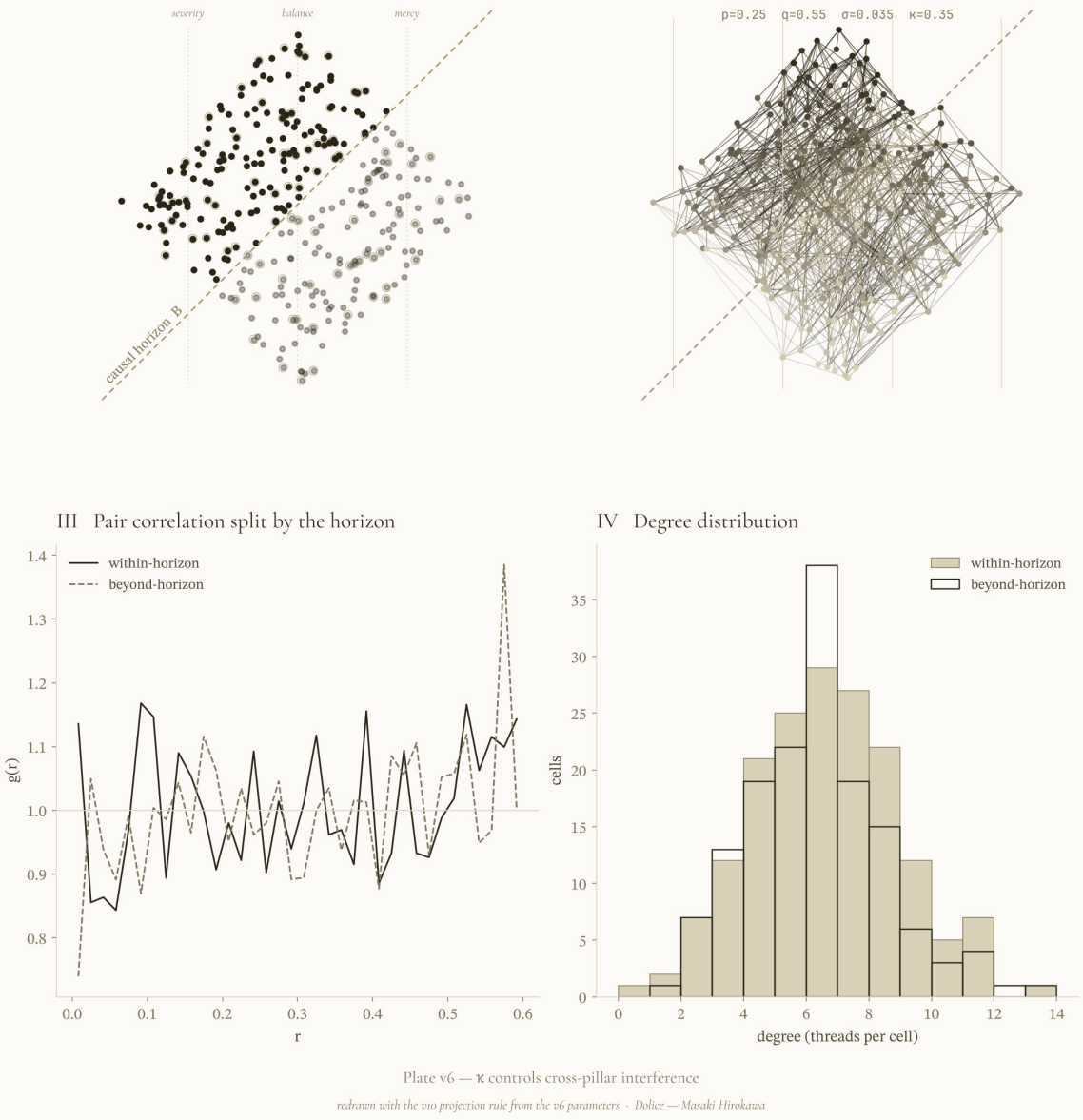

図 18 Plate v6 — κ controls cross-horizon interference

パラメータ	`p = 0.250  q = 0.550  σ = 0.035  κ = 0.350`
観測	地平跨ぎの混線が制御される
意図	κ で「破れ」を意図として扱う

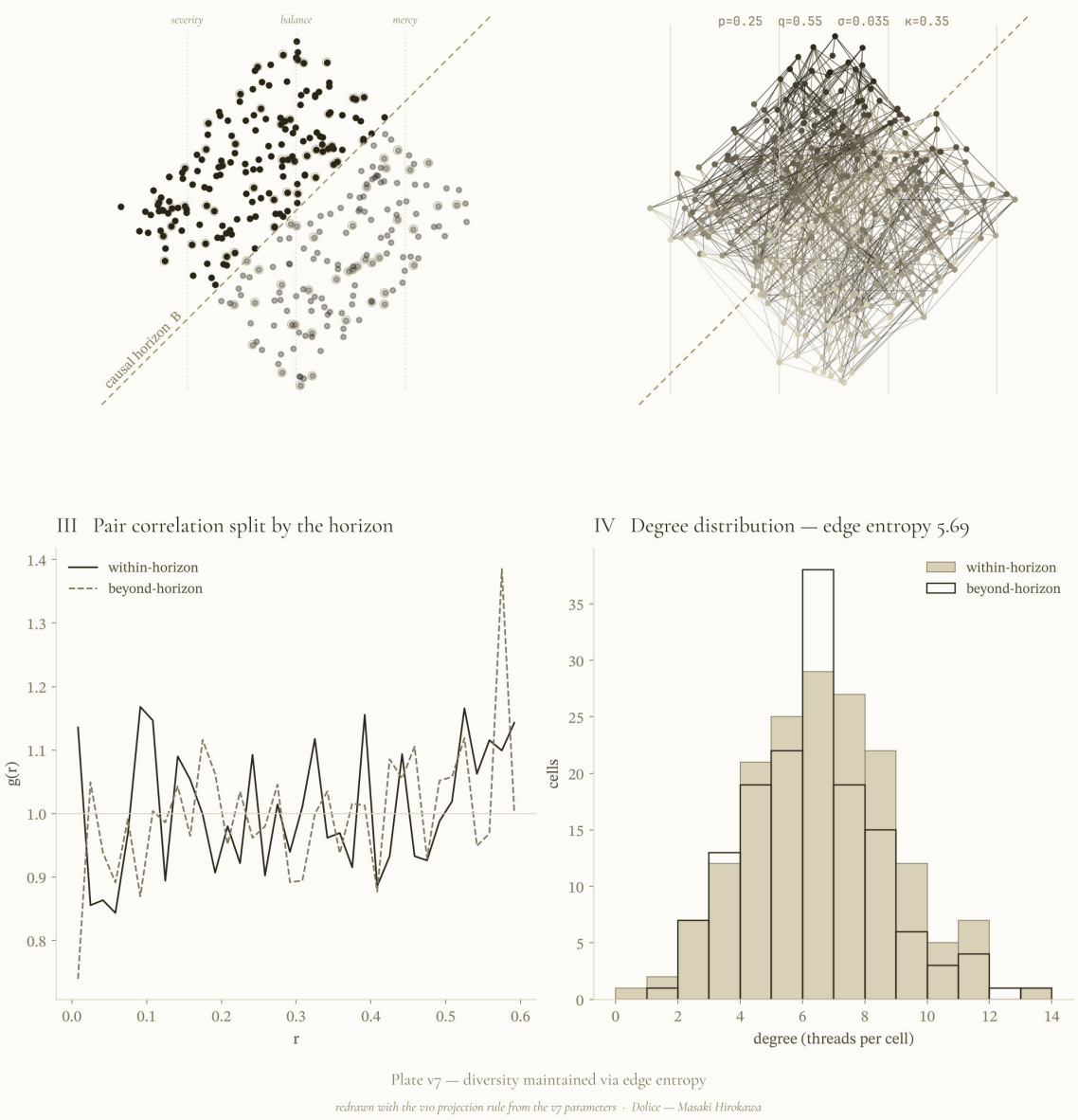

図 19 Plate v7 — diversity maintained via edge entropy

パラメータ	`p = 0.250  q = 0.550  σ = 0.035  κ = 0.350`
観測	単調化が起きにくい
意図	edge entropy で多様性を保持する

Plate v8 — Model → Output mapping clarified

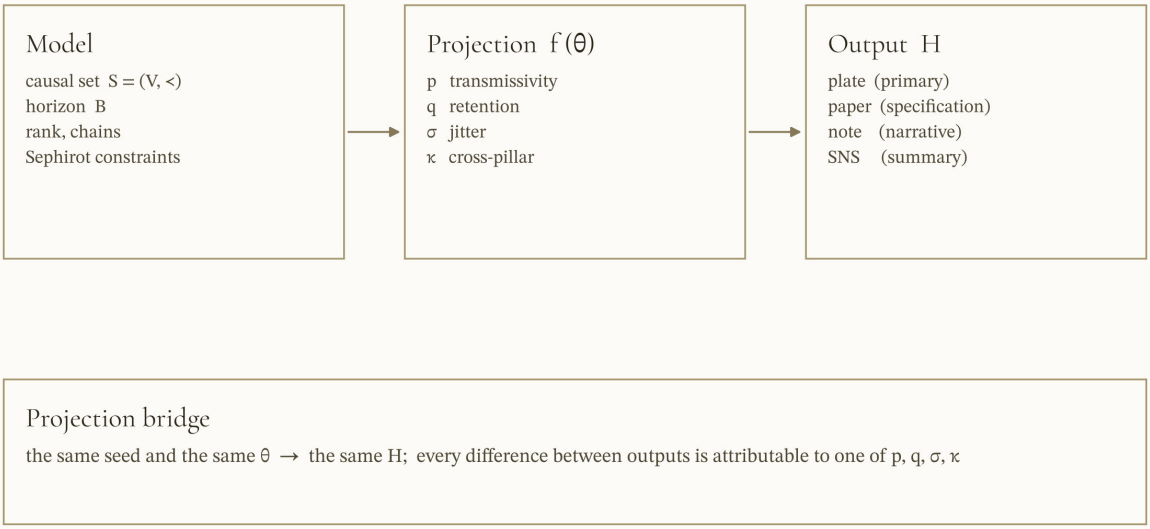

図 20 Plate v8 — Model → Output mapping clarified

パラメータ	`p = 0.500  q = 0.450  σ = 0.350  κ = 0.280`
観測	Model → Output の対応が整理される
意図	projection bridge を明文化する

Plate v8.1 — MVC remapping of the v7 specification

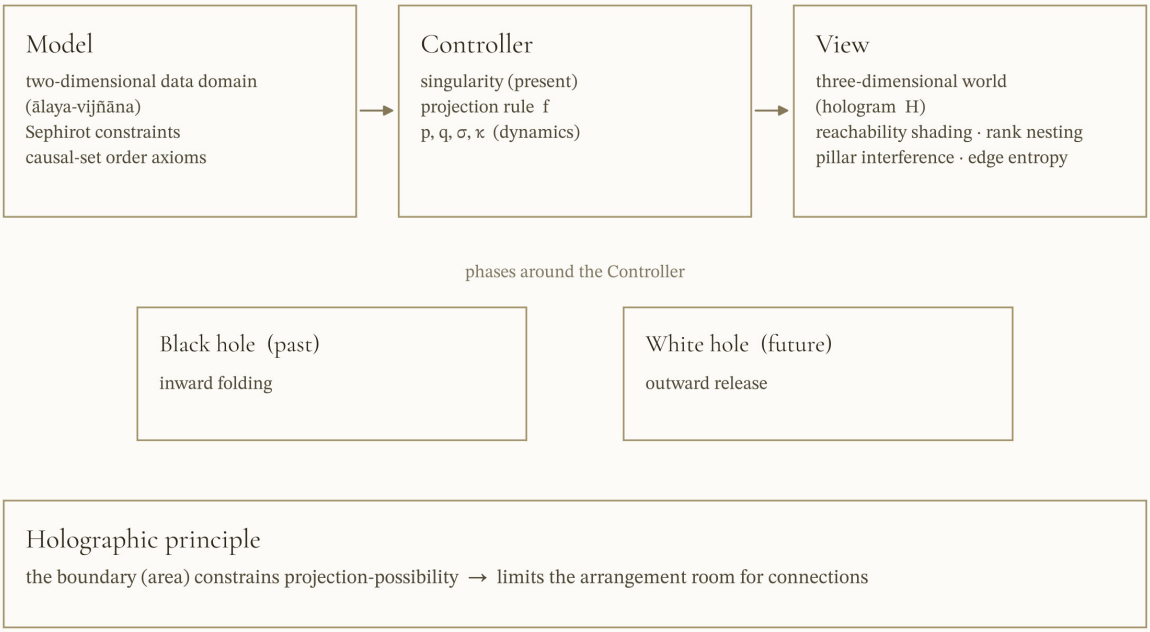

図 21 Plate v8.1 — interpretation stabilizes; MVC correspondence fixed

パラメータ	`p = 0.500  q = 0.450  σ = 0.350  κ = 0.280`
観測	解釈が安定する
意図	v7 を回収し、MVC 対応を固定する

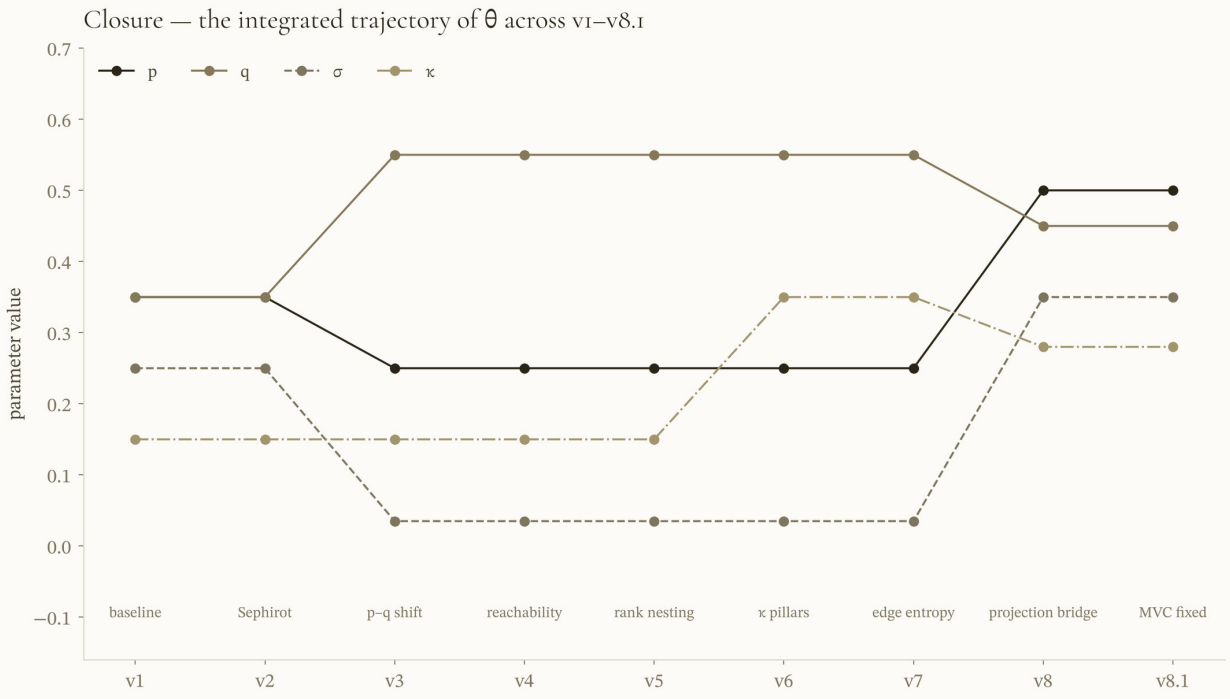

図 22 Closure — integrated trajectory (v1–v8.1)

パラメータ	`p = 0.500  q = 0.450  σ = 0.350  κ = 0.280`
観測	全体像を統合して提示する
意図	v1〜v8.1 の軌跡を俯瞰する



図 23 Plate v9 — projection rule f, stability domain, counterexamples

パラメータ	`p = 0.500  q = 0.450  σ = 0.350  κ = 0.280`
観測	公理・安定領域・反例が一枚に収まる
意図	f を五公理から再構成し、シリーズを閉じる

付録 A　コードの実行

本稿のコードは一つのファイル chfp.py（英語注釈版は chfp_en.py）にまとめてある。依存は numpy、networkx、matplotlib のみ。図版は figures.py が chfp.py を呼んで生成する。seed を変えなければ、本稿の数値と図はそのまま再現される。

実行

```
pip install numpy networkx matplotlib
python3 figures.py                      # every figure and plate → Figures/
python3 -c "import chfp as m; print(m.window_test())"
```

段階	関数	節
Model	sprinkle_diamond · causal_matrix · links · rank_longest_chain	§1
次元	ordering_fraction · myrheim_meyer_dimension	§1.2
地平面	horizon_split · cross_links · horizon_area	§2.1
投影 f	Controller · project · observe	§2.2–2.3
観測量	edge_entropy · reachability_depth · pair_correlation	§2.3
公理	check_axioms · is_interior · is_stable · phase_diagram	§4.5
境界制約	max_threads_across · kappa_max · area_law	§5.2, §5.4
弦	closed_string_spectrum · t_duality_holds	§6.2
曼荼羅・樹	tree_of_life · mandala_layout	§7
不生不滅	window_test · null_proper_time	§8.1–8.2
葬列・今・縁	minimum_chain_cover · chains_from_matching · maximum_antichain · common_future_fraction · en_window_test	§8.4–8.6
文化進化	moran_entropy	§9

付録 B　用語集

因果集合 / causal set	要素の集合と部分順序 $\prec$ の組。順序と個数だけで幾何を定める離散時空の模型。
散布 / sprinkling	時空の体積に比例した個数の点をポアソン過程で落とし、光円錐で順序を定める手続き。
リンク / link	中継のない因果関係。ハッセ図の辺。
rank	過去へ向かう最長鎖の長さ。葬列の段。
地平面 B / horizon	ヌル面 u = u_h。糸は外から内へしか越えられない。
θ = (p, q, σ, κ)	透過率・保持率・揺らぎ・干渉。投影規則 f の Controller。
Ω_stable	A1〜A5 が同時に成立する θ の集合。
ビット・スレッド / bit threads	境界を越える独立な糸の最大本数＝最小カット（Freedman–Headrick）。
Dou–Sorkin の数え方	地平面を横切るリンクの本数でブラックホールのエントロピーを数える提案。
Dilworth の定理	有限な部分順序は、幅に等しい本数の鎖に分割できる。
T 双対 / T-duality	半径 R の弦と α'/R の弦が同じスペクトルを持つこと。
反鎖 / antichain	互いに因果関係のない要素の集合。最大反鎖の大きさ＝幅＝葬列の本数（Dilworth）。本稿では「今」の定義。
Myrheim–Meyer 次元	関係のある対の割合 $f(d) = \Gamma(d+1)\Gamma(d/2)/(2\Gamma(3d/2))$ から次元 d を逆算する推定法。
縁（本稿）/ en	二つの要素に共通の未来が存在すること。$C\,C^{\mathsf{T}} > 0$。
窓の縁の定理（本稿）	極小元・極大元の割合は、窓を固定して領域を広げると 0 に向かう。

参考文献

L. Bombelli, J. Lee, D. Meyer, R. D. Sorkin, “Space-time as a causal set,” Phys. Rev. Lett. 59 (1987) 521.

R. D. Sorkin, “Causal sets: Discrete gravity,” in Lectures on Quantum Gravity (2005); arXiv:gr-qc/0309009.

D. Dou, R. D. Sorkin, “Black hole entropy as causal links,” Found. Phys. 33 (2003) 279; arXiv:gr-qc/0302009.

J. D. Bekenstein, “Black holes and entropy,” Phys. Rev. D 7 (1973) 2333. S. W. Hawking, “Particle creation by black holes,” Commun. Math. Phys. 43 (1975) 199.

S. Ryu, T. Takayanagi, “Holographic derivation of entanglement entropy from AdS/CFT,” Phys. Rev. Lett. 96 (2006) 181602.

M. Freedman, M. Headrick, “Bit threads and holographic entanglement,” Commun. Math. Phys. 352 (2017) 407; arXiv:1604.00354.

J. Maldacena, “The large N limit of superconformal field theories and supergravity,” Adv. Theor. Math. Phys. 2 (1998) 231.

E. Witten, “String theory dynamics in various dimensions,” Nucl. Phys. B 443 (1995) 85.

J. Polchinski, String Theory, Vols. I–II, Cambridge University Press (1998).（T 双対・D ブレーン）

R. P. Dilworth, “A decomposition theorem for partially ordered sets,” Ann. Math. 51 (1950) 161. D. R. Fulkerson, “Note on Dilworth's decomposition theorem,” Proc. AMS 7 (1956) 701.

J. Myrheim, “Statistical geometry,” CERN preprint TH-2538 (1978). D. A. Meyer, The Dimension of Causal Sets, PhD thesis, MIT (1988).

D. Kőnig, “Gráfok és mátrixok,” Matematikai és Fizikai Lapok 38 (1931) 116.

P. A. M. Moran, “Random processes in genetics,” Proc. Camb. Phil. Soc. 54 (1958) 60.

『般若波羅蜜多心経』（玄奘訳）「不生不滅 不垢不浄 不増不滅」。

M. Hirokawa, Causal Horizon Funeral Procession — Complete Edition (v9), Dolice (2026). https://dolice.net/pub/

Dolice

Causal Horizon Funeral Procession — Complete Edition v10

廣川政樹　Masaki Hirokawa

図版・数値はすべて付属のコードから再生成。作品はすべて AI を使用せず、作家自身の手で制作。

https://dolice.net　·　2026