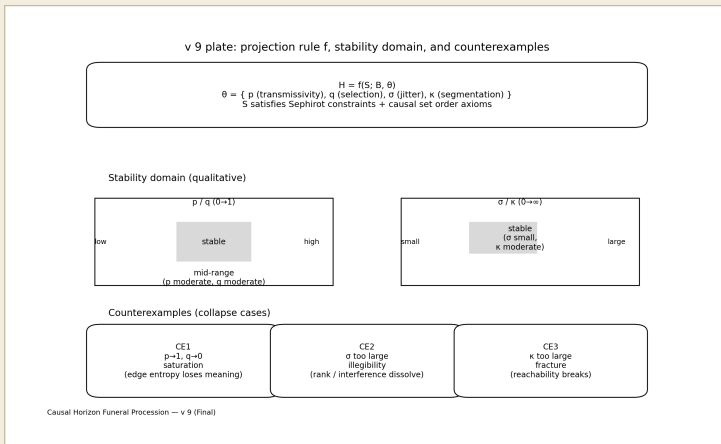


SEPHIROT HORIZON LAB

Causal Horizon Funeral Procession

葬 列

作品「Causal Sets / 葬列」を起点に、時間の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（v1～v9、投影規則 f の最小公理を含む）



v9 plate —— 投影規則 f 、安定領域、反例

廣川政樹 Masaki Hirokawa

COMPLETE EDITION • v1-v9 • JP

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	40 × 60 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

本作「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から着想を得ている。原因と結果の同時性を探求し、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直観によって、時間の連続性という前提に挑戦する。各存在はセルの中で孤立しているように描かれながらも、避けられない運命の絆によって互いに結ばれ、集合的な存在を形作っていく。星々の海を果てしなく続く比喩的な葬列を通じて、存在の儚さと、この移ろいやすい世界で一瞬一瞬やご縁を大切にすることの重要性を問いかける。

◆ 関連作（補助的参照）

「元型」の反復・入れ子の発想は、本研究における rank ネスト（v5）および反復構造の読解に、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が、rank ネストと反復構造の読解に補助的な参照枠を与える。

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

本作「元型」は「原型への回帰と反復」を主題とし、鳥の仮面を被った女性像の反復によって人間の根源的な欲求を象徴化する。作家にとって論理は感覚を抑える手段に過ぎず、その拘束が合わせ鏡のような終わりのない反復＝入れ子の世界を生む。この無限ループは、人類の歴史・文化・戦争・紛争、さらには宇宙の構築そのものを示唆する。

要旨

本稿は、作品「葬列」をブラックホール時空の地平面 B (horizon) 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 の更新は、(i) Sephirot 制約、(ii) 局所性／拡散の調整 (p, q, σ)、(iii) 跨柱干渉 (κ)、(iv) 接続多様性 (edge entropy)、(v) 投影規則 f の最小定義へと収束する。本稿は、これらを plate／論文／note の三層に一貫させる Protocol Mapping として提示し、再現可能な制作工程として固定する。完全版である本稿では、v9 において投影規則 f を五つの最小公理 (A1～A5) から再構成し、安定領域の代数的記述と一意性の主張をもってシリーズを閉じる。

1 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。

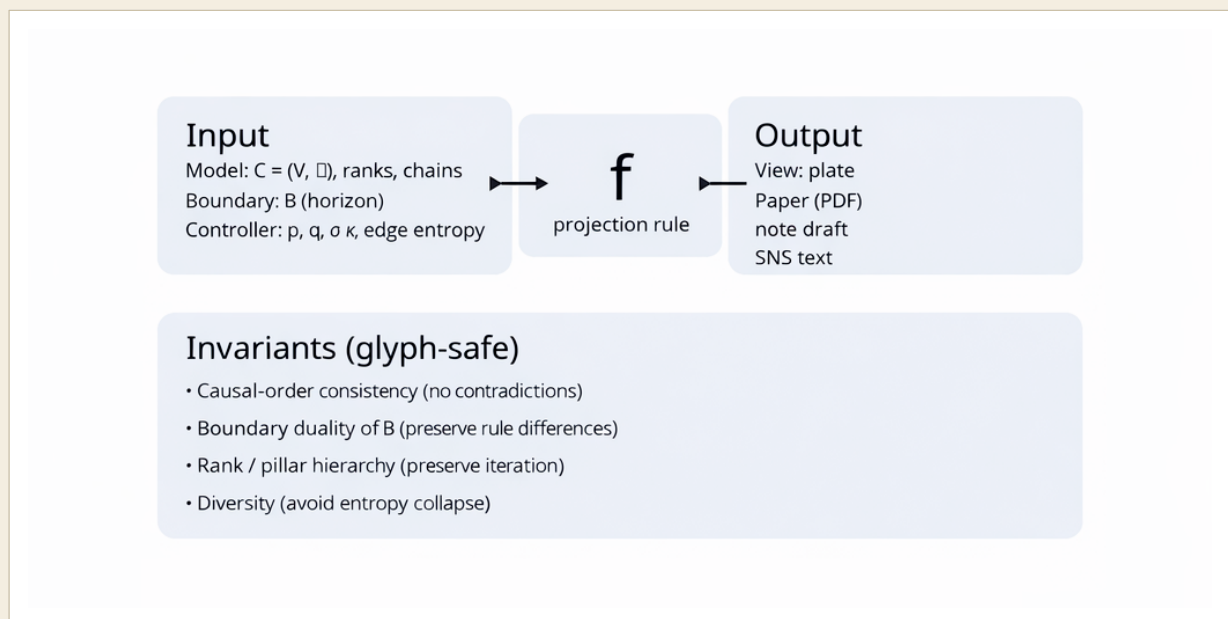


図 1：投影規則 f の入出力と不変量 (glyph-safe)。

◆ 内部状態（最小定義）

```
Input model : C = (V, <), rank, chains
Boundary    : B
Controller  : p, q,  $\sigma$ ,  $\kappa$ , entropy
Seed        : seed + protocol
```

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.gridspec import GridSpec

def knn_edges(pts, k=4):
    n = pts.shape[0]
    d2 = ((pts[:, None, :] - pts[None, :, :]) ** 2).sum(axis=2)
    np.fill_diagonal(d2, np.inf)
    idx = np.argpartition(d2, kth=k-1, axis=1)[:k]
    edges = set()
    for i in range(n):
        for j in idx[i]:
            j = int(j)
            a, b = (i, j) if i < j else (j, i)
            edges.add((a, b))
    return list(edges)

def pair_correlation(distances, r_max, bins=40):
    hist, ed = np.histogram(distances, bins=bins, range=(0, r_max))
    r = 0.5 * (ed[:-1] + ed[1:])
    c = hist / hist.max() if hist.max() > 0 else hist.astype(float)
    return r, c

def radial_power(points, grid=192):
    H, _, _ = np.histogram2d(points[:, 0], points[:, 1],
                              bins=grid, range=[[-1, 1], [-1, 1]])
    F = np.fft.fftshift(np.fft.fft2(H))
    P = np.abs(F) ** 2
    y, x = np.indices(P.shape)
    cy, cx = (np.array(P.shape) - 1) / 2
    rr = np.sqrt((x - cx) ** 2 + (y - cy) ** 2).astype(int)
    tbin = np.bincount(rr.ravel(), P.ravel())
    nr = np.bincount(rr.ravel())
    radial = tbin / np.maximum(nr, 1)
    k = np.arange(len(radial))
    return k, radial
```

◆ 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。plate では、horizon_y を境に上側（within-horizon）を構造的配置、下側（beyond-horizon）をランダム配置として生成し、点群と局所接続（kNN edges）を描画する。さらに、2 点相関を within-horizon/beyond-horizon/cross-horizon に分割して比較し、上側/下側の radial power（proxy）を併置する。

```

# requires: numpy as np, matplotlib.pyplot as plt, GridSpec
def render_plate(filename, seed=42, horizon_y=0.0, dpi=420):
    rng = np.random.default_rng(seed)

    # within-horizon (above): structured; beyond (below): random
    po = perturbed_hex_points(seed=seed)
    po = po[po[:, 1] ≥ horizon_y]
    pr = random_points(seed=seed + 1)
    pr = pr[pr[:, 1] < horizon_y]

    # ensure minimum count in the structured region
    if len(po) < 140:
        extra = po.copy(); extra[:, 0] *= -1
        po = np.vstack([po, extra]); po = po[po[:, 1] ≥ horizon_y]

    po = po[:180]; pr = pr[:180]
    pts = np.vstack([po, pr]); n = len(pts)
    edges = knn_edges(pts, k=4)

    above = set(np.where(pts[:, 1] ≥ horizon_y)[0].tolist())
    below = set(np.where(pts[:, 1] < horizon_y)[0].tolist())

    # correlation samples, split into within / beyond / cross-horizon
    # (full panel-drawing code in Appendix A)
    fig = plt.figure(figsize=(10, 10), constrained_layout=True)
    fig.suptitle("Causal Horizon Funeral Procession (monochrome plate)")
    fig.savefig(filename, dpi=dpi)
    plt.close(fig)

```

完全な実装は付録 A を参照。

2 不変量・反例・安定条件（中域）

不変量とは「壊すと作品として成立しない骨格」である。ここで中域とは、因果整合と地平の二相性を保ったまま、局所性／拡散（ p, q ）の調整が破綻へ転落しない範囲を指す。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（中域）の意味が具体化する。

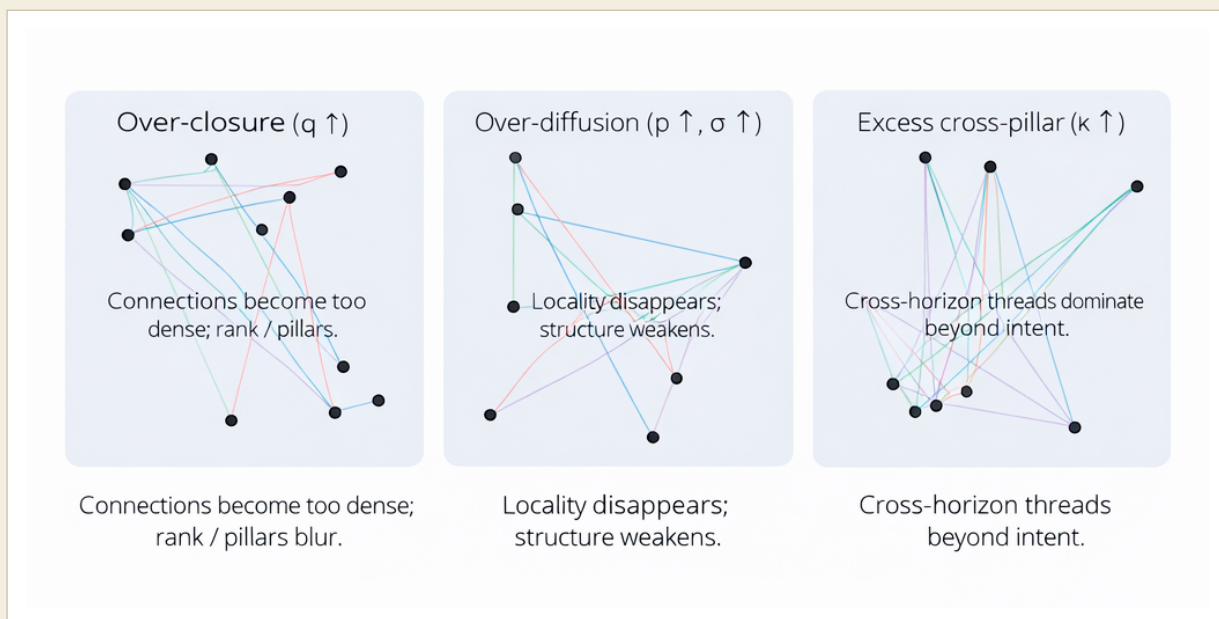


図 2：反例（破綻モード）。

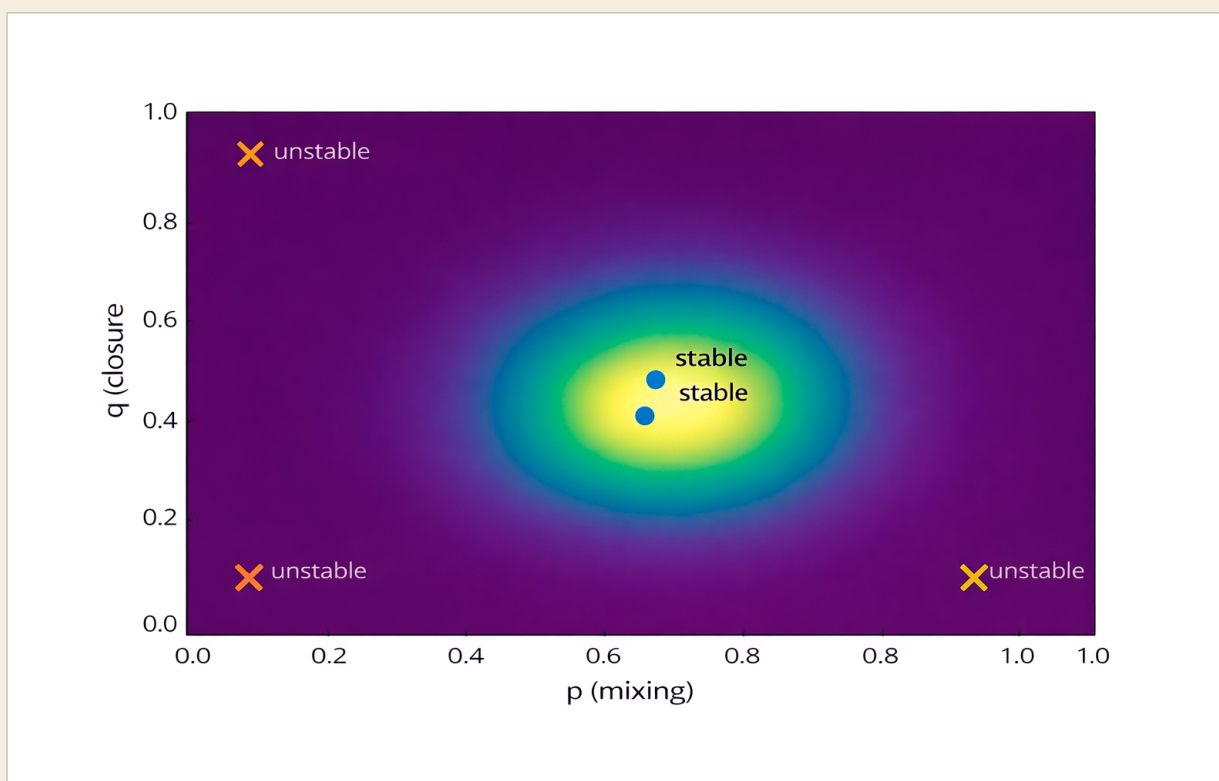


図 3：安定域（中域）の概念図（ p - q 平面）。

3 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。ここで同一の内部状態とは、同一 seed と Controller パラメータ集合 (p, q, σ, κ , edge entropy など) が保持された状態を指す。

4 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、ここでは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

v8.1 で固定された写像を次のように書く。「最小」とは「これを外すと葬列が成立しない」という意味であり、形式的な証明ではなく、反例による境界の確定によって示す——これは v1 から一貫して行われてきたことの明示化にほかならない。

$H = f(S; B, \theta)$	
$S = (V, <)$	causal set: vertices V with causal order $<$
B	causal horizon (boundary)
$\theta = \{ p, q, \sigma, \kappa, H_edge \}$	Controller parameters
H	View = Hologram (output)

◆ 4.2 五つの最小公理

A1 順序保存

$a < b \text{ in } S \Rightarrow f(a) \text{ precedes } f(b) \text{ in } H$

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)。ジッターが因果の糸を曖昧にし、どのセルがどのセルの「後」かが読み取れなくなる——判読不能 (illegibility)。

A2 地平面の二相性

B splits V into V_above (within) and V_below (beyond);
 $f(V_above)$ and $f(V_below)$ are structurally distinct.

$p \rightarrow 1$ のとき within と beyond が混合し、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。

違反するとき： $p \rightarrow 1$, $q \rightarrow 0$ (CE1)。飽和 (saturation) は、地平面の意味喪失と edge entropy の崩壊が同時に起きる状態である。

A3 局所連続性

for small $\delta\theta$, the change δH is bounded and continuous;
 f is Lipschitz in θ within the stable regime.

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。

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

$H_{\text{edge}}(f(S; B, \theta)) \geq H_{\text{min}} > 0$

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 (均質化による entropy 崩壊)。

A5 ホログラフィック境界制約

$|\text{cross-horizon edges}| \leq \text{Area}(B) / \ell_P^2$

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。

違反するとき： κ が大きすぎる (CE3)。Cross-horizon threads が多すぎると reachability が破れ、葬列の「列」構造が断裂 (fracture) する——柱 (pillar) の意味が消える。

◆ 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	$p \rightarrow 1$, $q \rightarrow 0$	A2, A4	地平面消失 + entropy 崩壊
CE2	σ 大	A1	因果順序の読み取り不能 (illegibility)
CE3	κ 大	A5	到達可能性の破損 (fracture)

A3 (局所連続性) は CE1～CE3 のすべての境界で破れる。安定領域の外縁が相転移点であることの別の言い方である。

◆ 4.4 安定領域の代数的記述

五公理が同時に成立するパラメータの集合 Ω_{stable} を次のように定義する。v8.1 の観測「安定領域は p/q 中域、 σ 小、 κ 中程度」は、この代数的記述の経験的確認であった。

```
 $\Omega_{\text{stable}} = \{ \theta = (p, q, \sigma, \kappa) :$   
  A1:  $\sigma < \sigma_{\text{max}}(q)$              $\sigma$  cap depends on  $q$  (selection supports order)  
  A2:  $p < p_{\text{max}}, q > q_{\text{min}}$      $p$  not too large,  $q$  not too small  
  A3:  $\theta \in \text{int}(\Omega_{\text{stable}})$         interior only (the boundary is non-continuous)  
  A4:  $H_{\text{edge}}(\theta) \geq H_{\text{min}}$         entropy floor  
  A5:  $\kappa < \kappa_{\text{max}}(p, q)$          $\kappa$  cap depends on  $p, q$  }
```

◆ 4.5 公理の検証関数

各公理を個別の関数として実装し、与えられたパラメータ θ が安定領域に属するかを検査する。閾値は v1~v8.1 の経験から定めた近似であり、今後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暫定値である。

```
import math  
from dataclasses import dataclass  
  
@dataclass  
class Controller:  
    p: float        # transmissivity / mixing    (0, 1)  
    q: float        # selection / closure        (0, 1)  
    sigma: float    # jitter                    ( $\geq 0$ )  
    kappa: float    # segmentation / cross      ( $\geq 0$ )  
  
def check_axioms(theta: Controller) → dict:  
    """Verify whether a Controller theta satisfies each axiom A1-A5."""  
    p, q, sigma, kappa = theta.p, theta.q, theta.sigma, theta.kappa  
  
    # A1 Order preservation: small jitter keeps the causal order legible  
    sigma_max = 0.15 * (q + 0.1)  
    a1 = sigma < sigma_max  
  
    # A2 Boundary duality: the horizon must keep two distinct phases  
    a2 = (p < 0.85) and (q > 0.15)  
  
    # A3 Local continuity: theta sits in the interior of the stable set  
    dist = min(abs(p - 0.2), abs(p - 0.8), abs(q - 0.2), abs(q - 0.7))  
    a3 = dist > 0.05  
  
    # A4 Entropy non-collapse: edge entropy stays above a floor  
    h = -p * math.log(p + 1e-9) - (1 - p) * math.log(1 - p + 1e-9)  
    h *= q  
    a4 = h > 0.1  
  
    # A5 Holographic bound: cross-horizon threads stay under the area cap  
    kappa_max = 0.5 * (1 - p) * q  
    a5 = kappa < kappa_max  
  
    return {"A1": a1, "A2": a2, "A3": a3, "A4": a4, "A5": a5}
```

```
def is_stable(theta: Controller) → bool:
    return all(check_axioms(theta).values())
```

◆ 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。アートの
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 $S, B, \theta \in \Omega_{\text{stable}}$ に対して、 $f(S; B, \theta)$ 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。

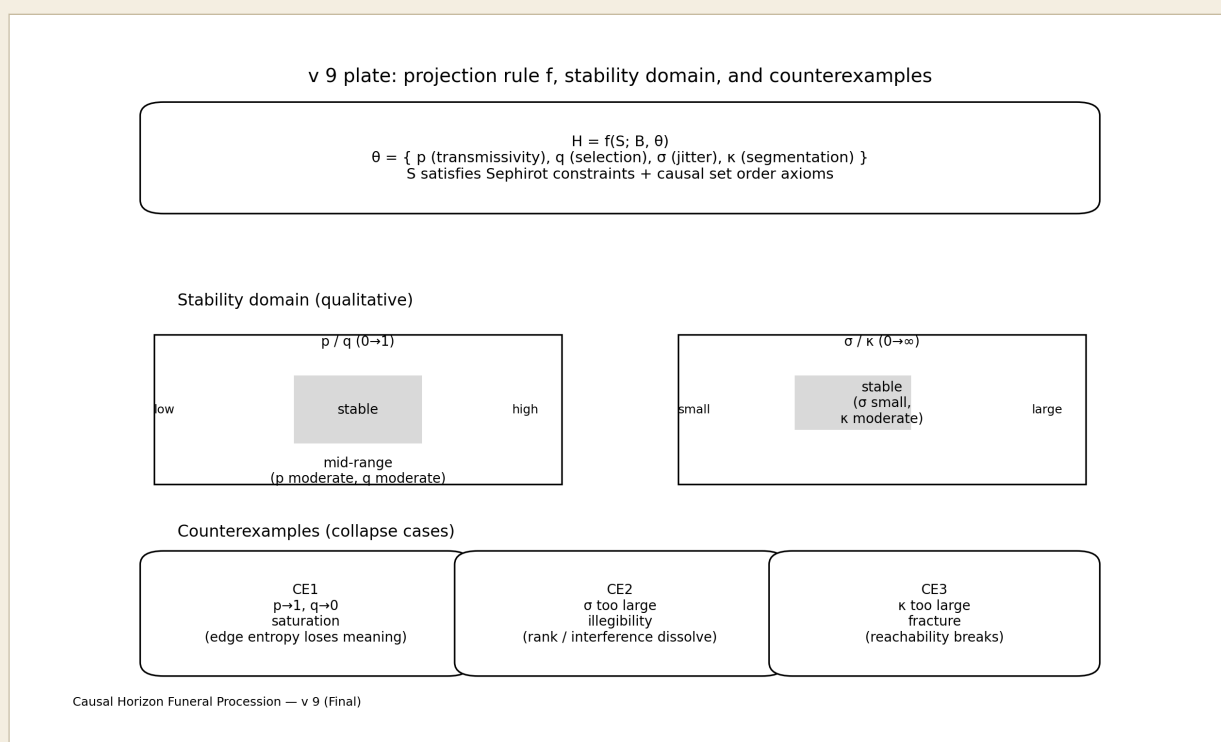


図 14：v9 plate —— 投影規則 f 、安定領域、反例（CE1～CE3）。

◆ 4.7 v9 が閉じるもの

v9 が完成したとき、「葬列」は次の意味を持つ。「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。」v1 で「原因と結果の同時性」「存在の儚さ」「一瞬一瞬のご縁」と書いたことは、数学的には「A1～A5 の交叉が定義する安定領域の中でのみ意味を持つ接続の

束」として記述できる。詩と数学が同じことを言っている——それが v1 から追いかけていたものである。

5 plate ギャラリー (v1~v8.1 + closure)

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。「観測」は plate 上で読み取れる構造変化を、「意図」は当該パラメータ設定が担う設計目的を指す。

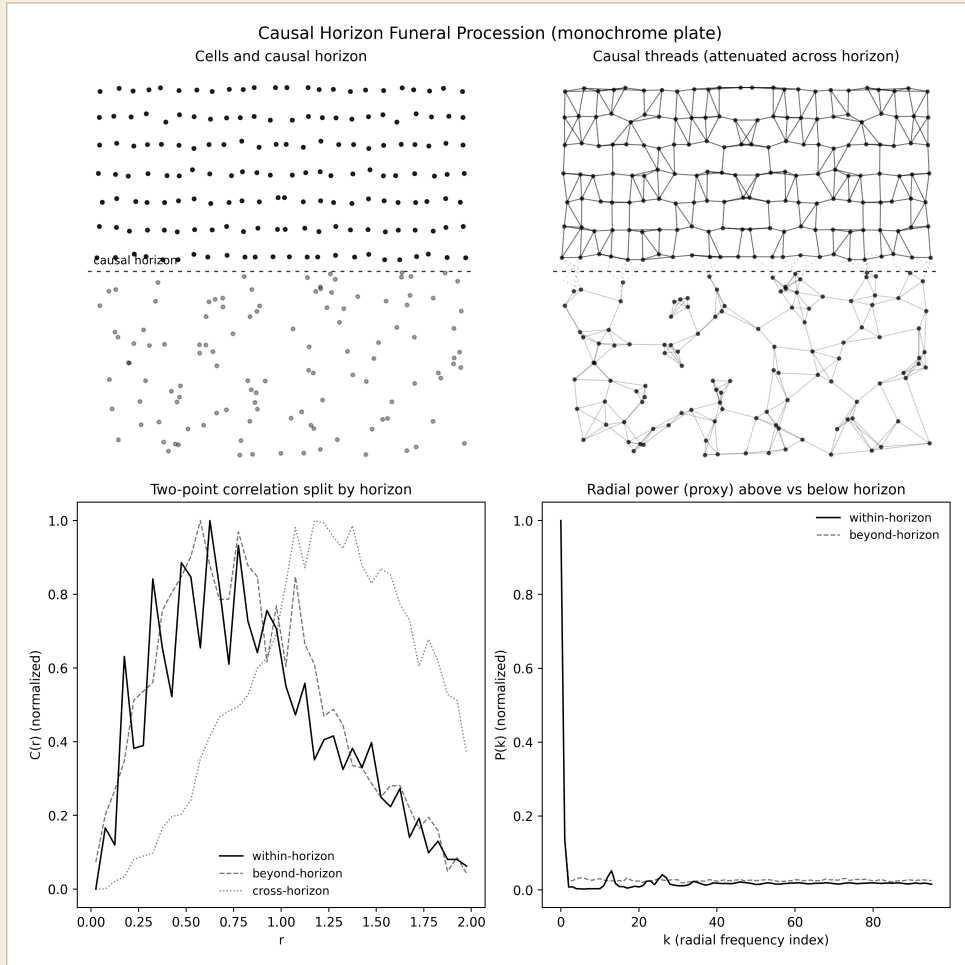


図 4 : Plate v1 (baseline) — two-phase split across the horizon

$$p = 0.350 \quad q = 0.350 \quad \sigma = 0.250 \quad \kappa = 0.150$$

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

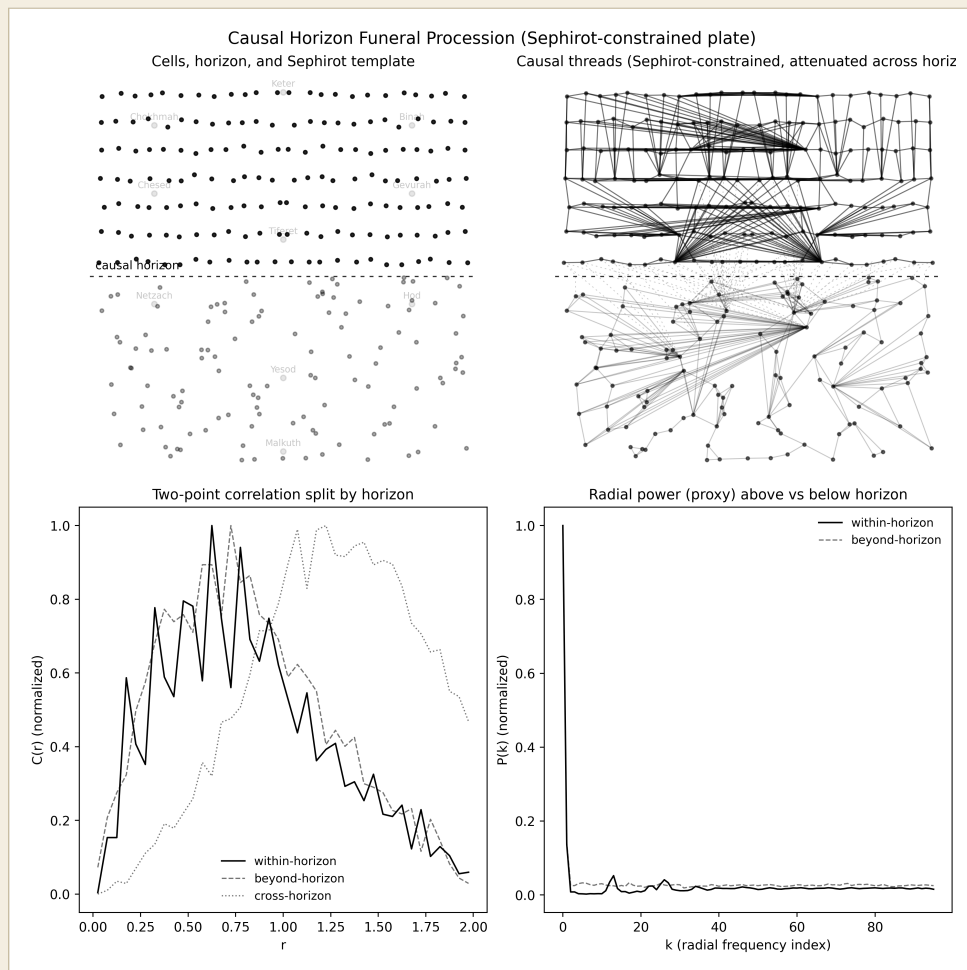


図 5 : Plate v2 — coherent density and increased order under constraints

$$p = 0.350 \quad q = 0.350 \quad \sigma = 0.250 \quad \kappa = 0.150$$

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

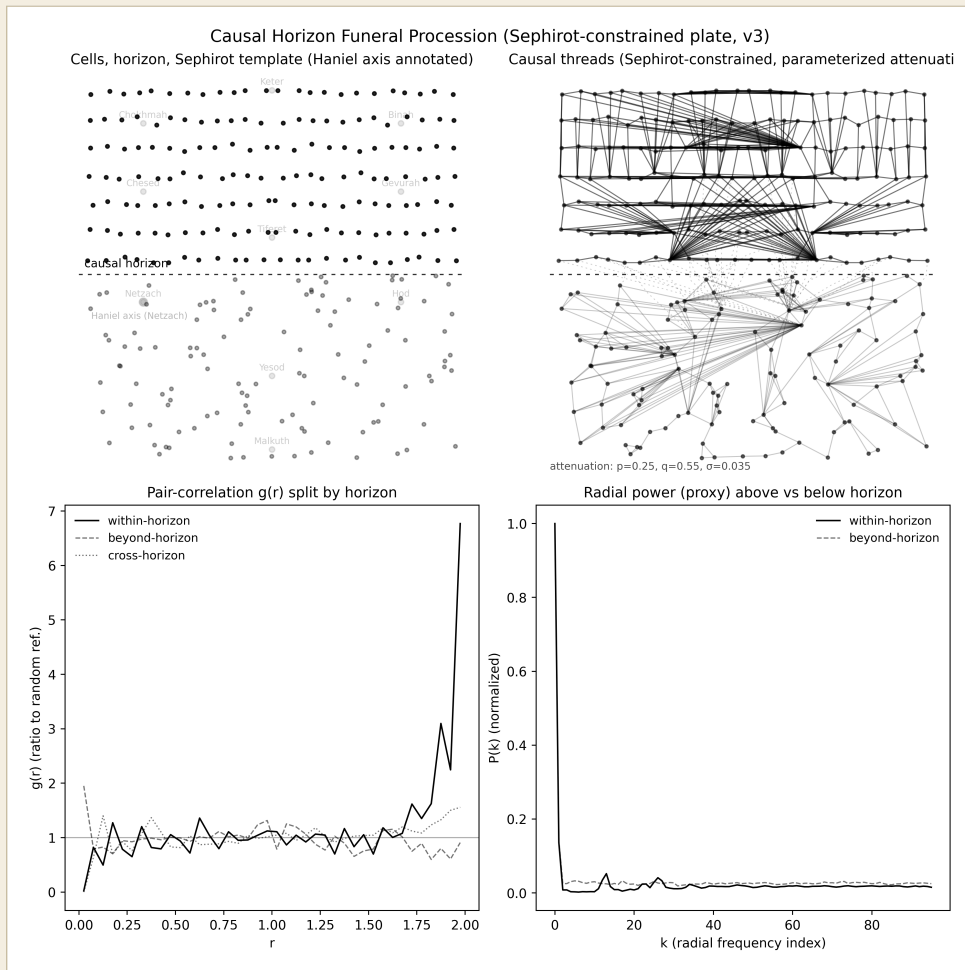


図 6 : Plate v3 — locality and haze in balance (p-q shift)

$p = 0.250$ $q = 0.550$ $\sigma = 0.035$ $\kappa = 0.150$

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

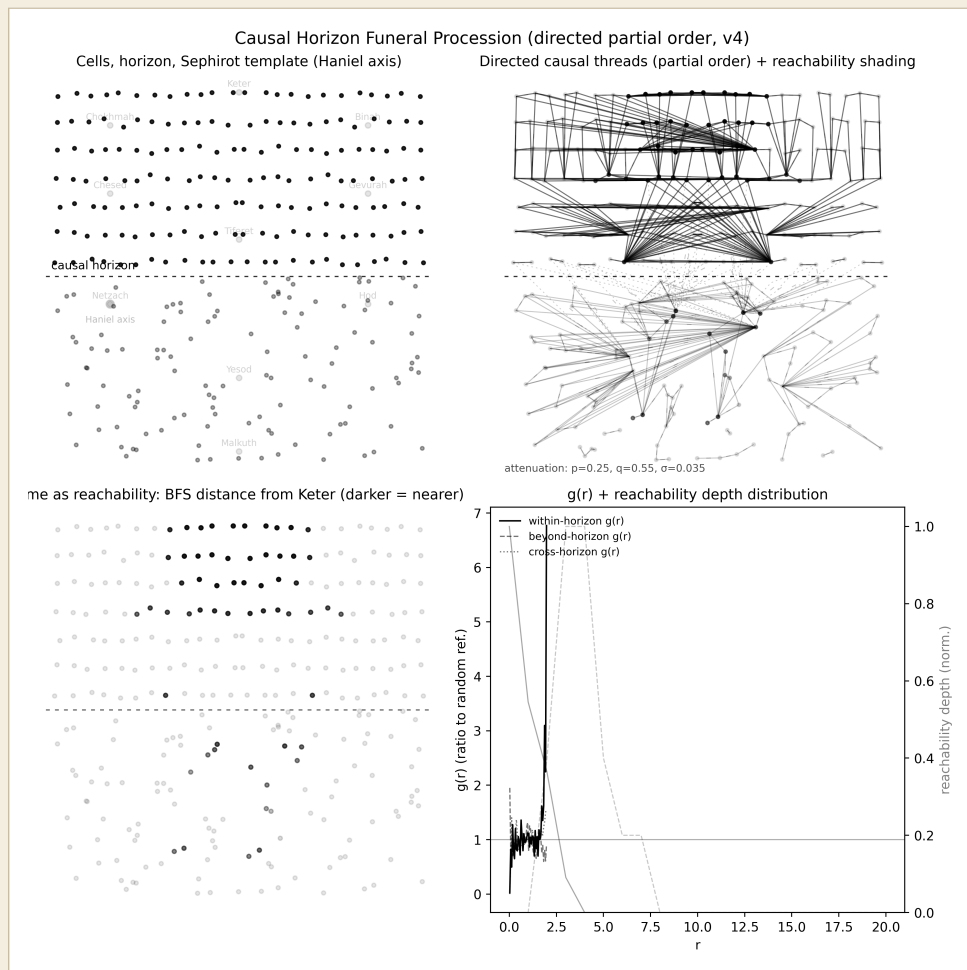


図 7 : Plate v4 — reachability becomes more legible

$$p = 0.250 \quad q = 0.550 \quad \sigma = 0.035 \quad \kappa = 0.150$$

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

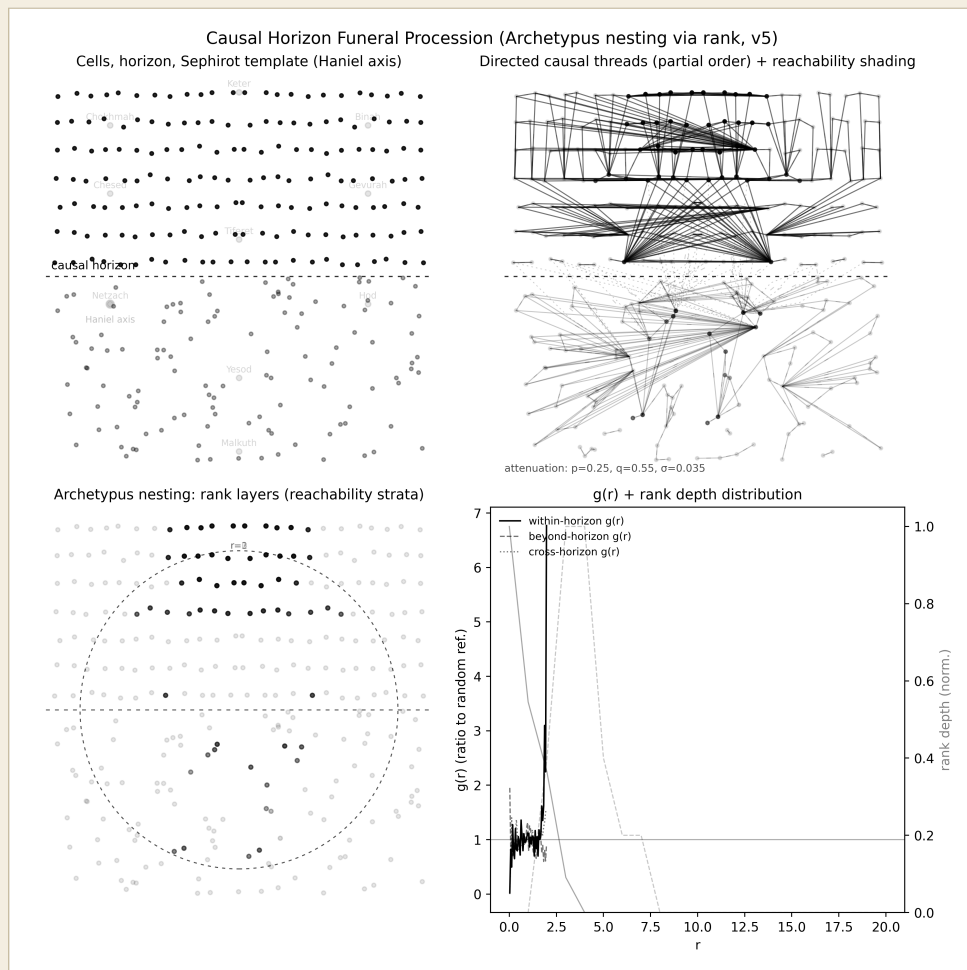


図 8 : Plate v5 — rank repetition becomes prominent

$$p = 0.250 \quad q = 0.550 \quad \sigma = 0.035 \quad \kappa = 0.150$$

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

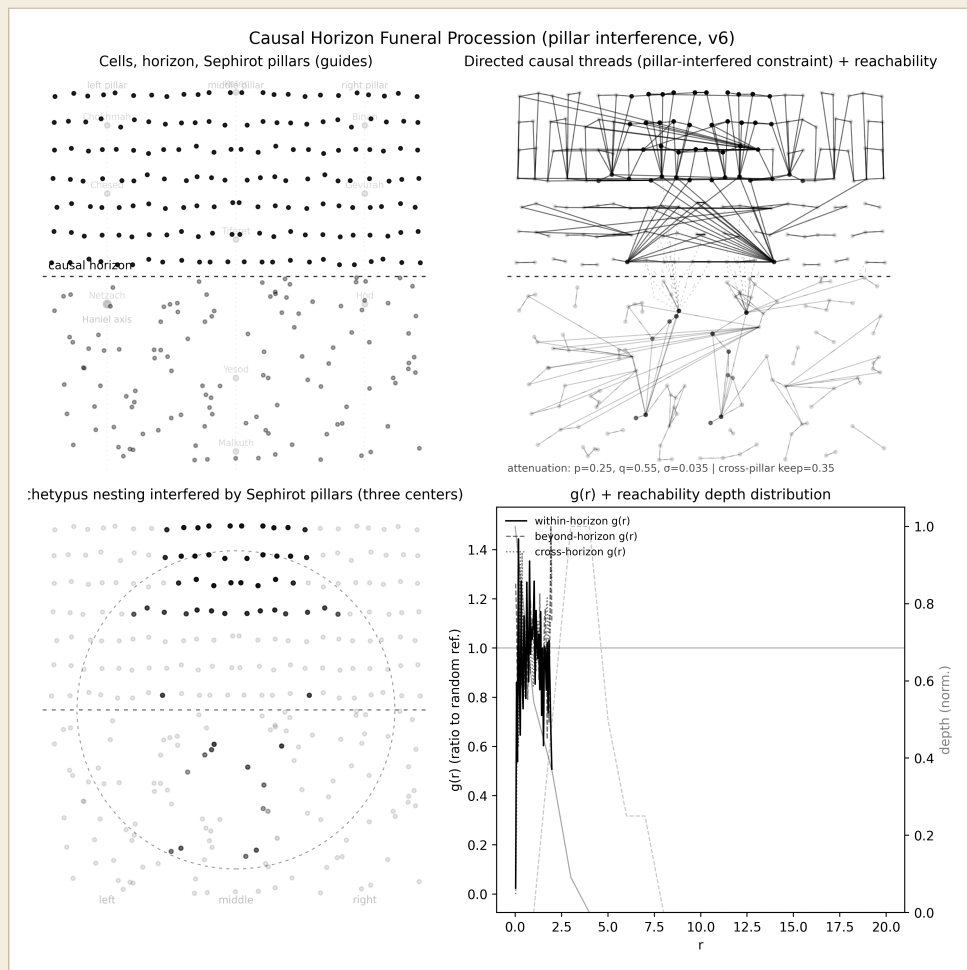


図 9 : Plate v6 — κ controls cross-horizon interference

$$p = 0.250 \quad q = 0.550 \quad \sigma = 0.035 \quad \kappa = 0.350$$

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

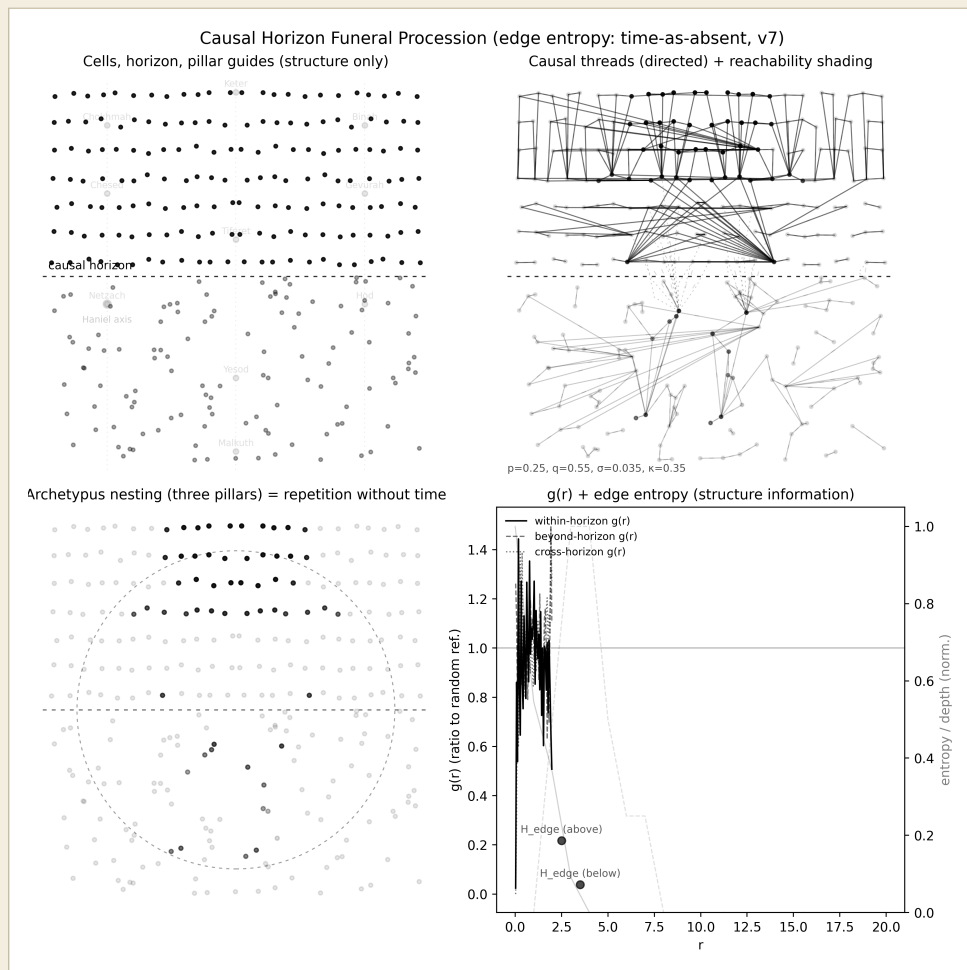


図 10 : Plate v7 — diversity maintained via edge entropy

$$p = 0.250 \quad q = 0.550 \quad \sigma = 0.035 \quad \kappa = 0.350$$

観測 単調化が起きにくい

意図 edge entropy で多様性を保持する

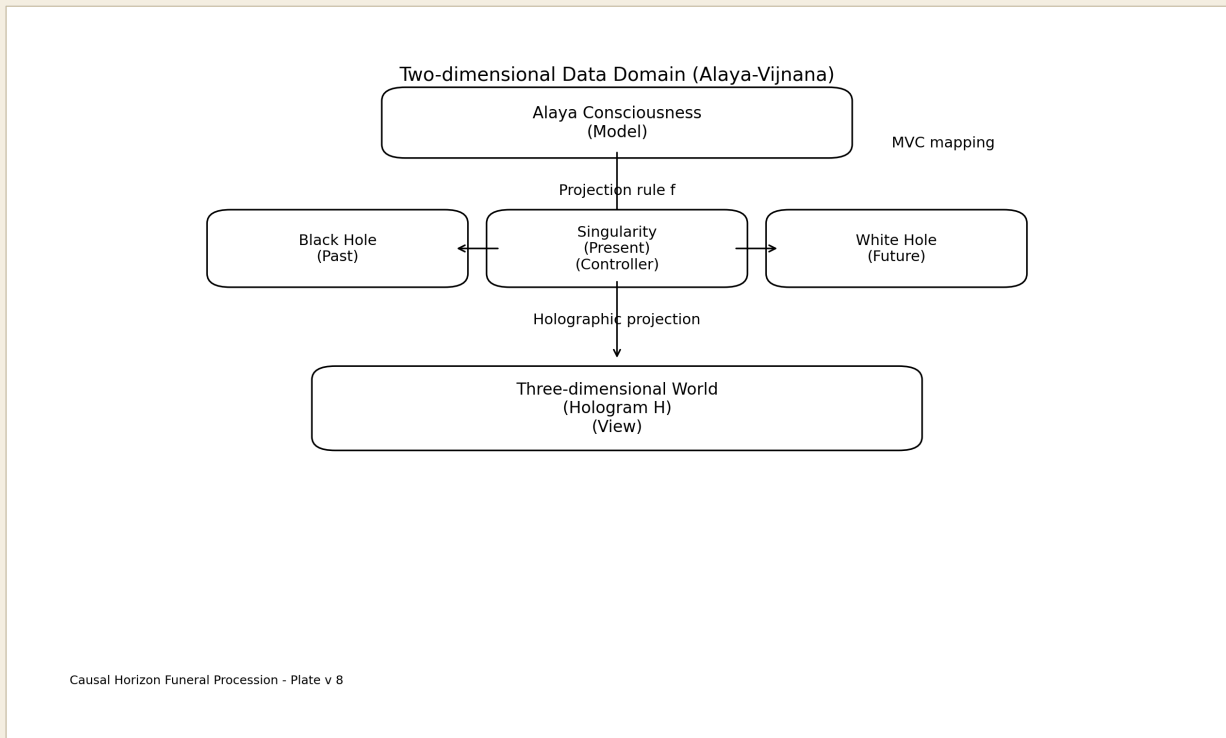


図 11 : Plate v8 – Model → Output mapping clarified

$$p = 0.500 \quad q = 0.450 \quad \sigma = 0.350 \quad \kappa = 0.280$$

観測 Model → Output の対応が整理される

意図 projection bridge を明文化する

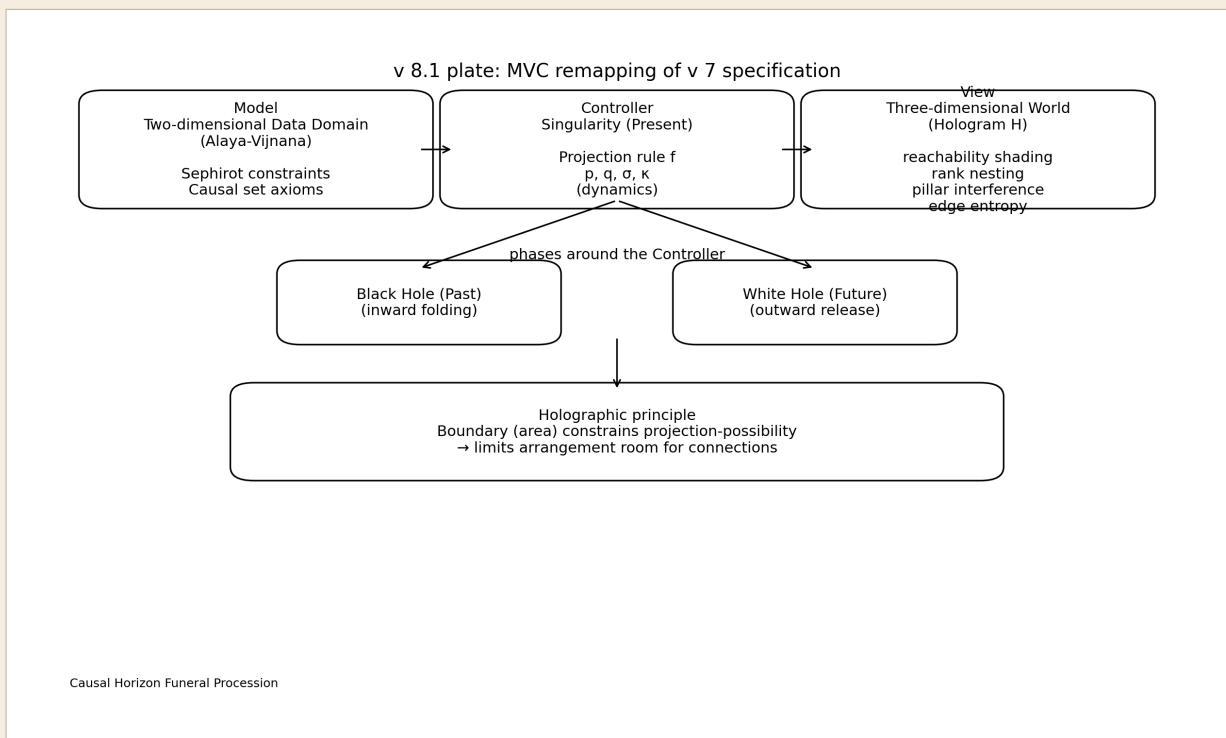


図 12 : Plate v8.1 – interpretation stabilizes; MVC correspondence fixed

$$p = 0.500 \quad q = 0.450 \quad \sigma = 0.350 \quad \kappa = 0.280$$

観測 解釈が安定する

意図 v7 を回収し、MVC 対応を固定する

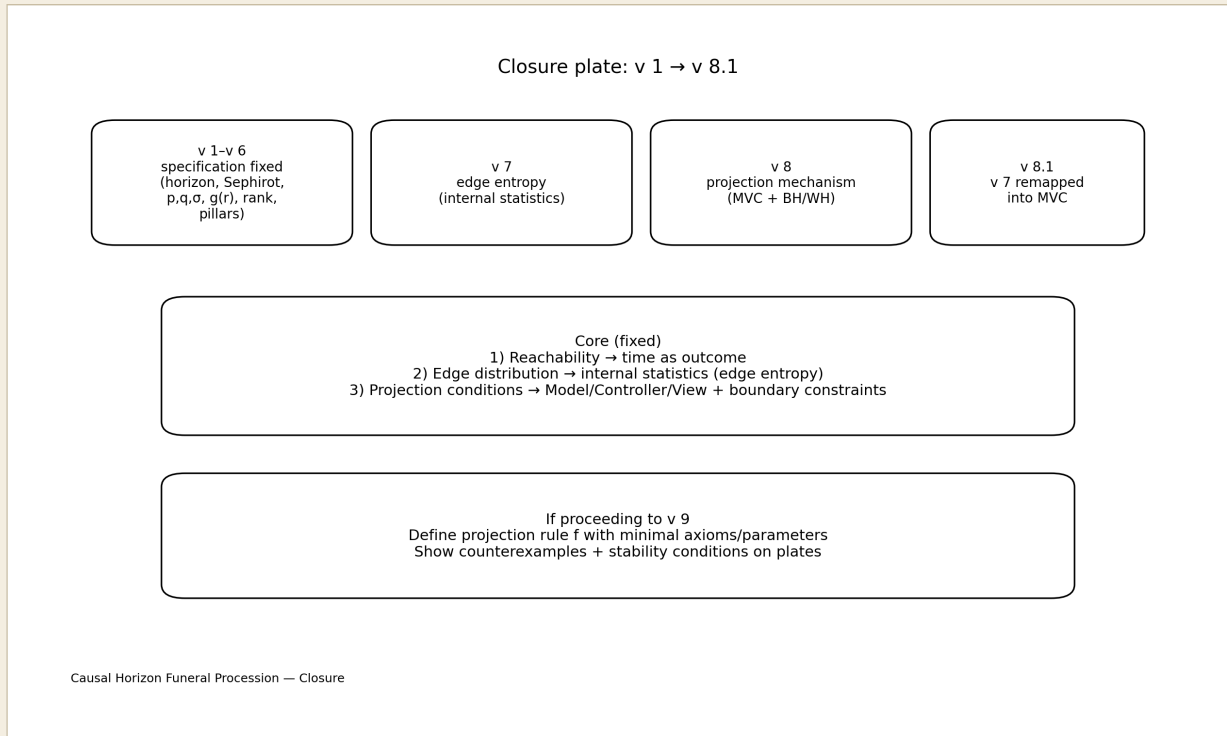


図 13 : Closure — integrated trajectory (v1-v8.1)

$$p = 0.500 \quad q = 0.450 \quad \sigma = 0.350 \quad \kappa = 0.280$$

観測 全体像を統合して提示する

意図 締めとして v1~v8.1 の軌跡を俯瞰する

結語

本完全版は、作品「葬列」を起点とする v1~v9 の軌跡を、一つの公理系のもとに閉じる試みである。plate（視覚）、論文（仕様）、note（語り）、SNS（共有）の四出力は、同一の内部状態から派生する一つの構造の異なる表面であった。投影規則 f が五つの公理から一意に定まるという主張は、証明の完成というより、これからの問いの出発点である—— $\kappa_{\max}$ のホログラフィック導出、境界における f の挙動、Sephirot 制約の公理化、そして文化進化論との接続（A4 = 多様性の保全）。これらは、いつか続きを書く日のために開かれたままにしておく。

付録 A 参照コード (render_plate)

§1 で要約した plate 生成の完全な実装を以下に示す。本文の helper 関数 (knn_edges, pair_correlation, radial_power) と同一の内部状態 (seed・パラメータ集合) を前提とする。

```
def render_plate(filename, seed=42, horizon_y=0.0, dpi=420):
    rng = np.random.default_rng(seed)

    po = perturbed_hex_points(seed=seed)
    po = po[po[:, 1] ≥ horizon_y]
    pr = random_points(seed=seed + 1)
    pr = pr[pr[:, 1] < horizon_y]

    if len(po) < 140:
        extra = po.copy(); extra[:, 0] *= -1
        po = np.vstack([po, extra]); po = po[po[:, 1] ≥ horizon_y]

    po = po[:180]; pr = pr[:180]
    pts = np.vstack([po, pr]); n = len(pts)
    edges = knn_edges(pts, k=4)
    above = set(np.where(pts[:, 1] ≥ horizon_y)[0].tolist())
    below = set(np.where(pts[:, 1] < horizon_y)[0].tolist())

    pair_samples = 20000
    a = rng.integers(0, n, size=pair_samples)
    b = rng.integers(0, n, size=pair_samples)
    m = a ≠ b; a = a[m]; b = b[m]
    dist = np.sqrt(((pts[a] - pts[b]) ** 2).sum(axis=1))

    dA, dB, dC = [], [], []
    for i in range(len(dist)):
        ia, ib = int(a[i]), int(b[i])
        if ia in above and ib in above: dA.append(dist[i])
        elif ia in below and ib in below: dB.append(dist[i])
        else: dC.append(dist[i])

    rA, cA = pair_correlation(np.array(dA), r_max=2.0, bins=40)
    rB, cB = pair_correlation(np.array(dB), r_max=2.0, bins=40)
    rC, cC = pair_correlation(np.array(dC), r_max=2.0, bins=40)
    kA, pA = radial_power(po, grid=192)
    kB, pB = radial_power(pr, grid=192)

    fig = plt.figure(figsize=(10, 10), constrained_layout=True)
    gs = GridSpec(2, 2, figure=fig)

    ax1 = fig.add_subplot(gs[0, 0]); ax1.set_aspect("equal"); ax1.axis("off")
    ax1.scatter(po[:, 0], po[:, 1], s=8, color="black", alpha=0.85)
    ax1.scatter(pr[:, 0], pr[:, 1], s=8, color="black", alpha=0.35)
    ax1.plot([-1.05, 1.05], [horizon_y, horizon_y],
            linestyle=(0, (3, 4)), linewidth=1.0, color="black", alpha=0.8)
    ax1.text(-1.02, horizon_y + 0.04, "causal horizon", fontsize=9)
    ax1.set_xlim(-1.1, 1.1); ax1.set_ylim(-1.1, 1.1)
    ax1.set_title("Cells and causal horizon", fontsize=11)

    ax2 = fig.add_subplot(gs[0, 1]); ax2.set_aspect("equal"); ax2.axis("off")
    ax2.plot([-1.05, 1.05], [horizon_y, horizon_y],
            linestyle=(0, (3, 4)), linewidth=1.0, color="black", alpha=0.8)
```

```

for (i, j) in edges:
    xi, yi = pts[i]; xj, yj = pts[j]
    crosses = (yi - horizon_y) * (yj - horizon_y) < 0
    if crosses:
        mx, my = 0.5 * (xi + xj), 0.5 * (yi + yj)
        jx = mx + rng.normal(0, 0.03); jy = my + rng.normal(0, 0.03)
        ax2.plot([xi, jx, xj], [yi, jy, yj], linewidth=0.6,
                  alpha=0.25, color="black", linestyle=(0, (2, 4)))
    else:
        if yi ≥ horizon_y and yj ≥ horizon_y:
            ax2.plot([xi, xj], [yi, yj], linewidth=0.8, alpha=0.55, color="black")
        else:
            ax2.plot([xi, xj], [yi, yj], linewidth=0.7, alpha=0.22, color="black")
ax2.scatter(pts[:, 0], pts[:, 1], s=6, color="black", alpha=0.6)
ax2.set_xlim(-1.1, 1.1); ax2.set_ylim(-1.1, 1.1)
ax2.set_title("Causal threads (attenuated across horizon)", fontsize=11)

ax3 = fig.add_subplot(gs[1, 0])
ax3.plot(rA, cA, linewidth=1.2, color="black")
ax3.plot(rB, cB, linewidth=1.0, color="black", alpha=0.55, linestyle="--")
ax3.plot(rC, cC, linewidth=1.0, color="black", alpha=0.55, linestyle=":")
ax3.set_xlabel("r"); ax3.set_ylabel("C(r) (normalized)")
ax3.set_title("Two-point correlation split by horizon", fontsize=11)
ax3.legend(["within-horizon", "beyond-horizon", "cross-horizon"],
           frameon=False, fontsize=9)

ax4 = fig.add_subplot(gs[1, 1])
ax4.plot(kA, pA, linewidth=1.2, color="black")
ax4.plot(kB, pB, linewidth=1.0, color="black", alpha=0.55, linestyle="--")
ax4.set_xlabel("k (radial frequency index)"); ax4.set_ylabel("P(k) (normalized)")
ax4.set_title("Radial power (proxy) above vs below horizon", fontsize=11)
ax4.legend(["within-horizon", "beyond-horizon"], frameon=False, fontsize=9)

fig.suptitle("Causal Horizon Funeral Procession (monochrome plate)", fontsize=13)
fig.savefig(filename, dpi=dpi)
plt.close(fig)

```