In[42]:=
```
(* ============================================== *)
(* COMPUTATIONAL UNIVERSE: BLACK HOLE HOLOGRAPHY     *)
(* 任务：验证 AdS 神经元时空的面积-信息关系 (S ~ A)      *)
(* 机制：严格执行 活性(表面) vs 冻结(内部) 分离          *)
(* ============================================== *)

ClearAll[];
SeedRandom[];

(* 1. 严格演化引擎 (Strict Evolution Engine) *)
(* 只有无向边(Active)能反应，反应后变为有向边(Inert) *)
StepEvolution[g_Graph] := Module[{
  allEdges, activeEdges, candidates, e1, e2,
  x, y, z, w, maxNode,
  newActive, newInert, remainingEdges
},
  allEdges = EdgeList[g];
  (* 提取活性边界： 只有无向边是活的 *)
  activeEdges = Cases[allEdges, _UndirectedEdge];

  If[Length[activeEdges] < 2, Return[g]];

  (* 随机选择反应对： 模拟量子涨落 *)
  candidates = RandomSample[activeEdges];

  (* 寻找共享顶点的对 {{x,y}, {y,z}} *)
  (* 这里不强制 Window，依靠活性边的边缘分布自然形成 AdS *)
  Do[
    e1 = candidates[[i]];
    (* 简单起见，只向后搜索少量邻居，维持一定的局部性 *)
    Do[
      e2 = candidates[[j]];
      If[Length[Intersection[List @@ e1, List @@ e2]] == 1,
        Goto[]
      ],
      {j, i + 1, Min[i + 10, Length[candidates]]}
    ],
    {i, 1, Length[candidates]}
  ];
  Return[g]; (* 未找到则跳过 *)
```

```
  Label[];

  (* 确定拓扑位置 *)
  y = Intersection[List @@ e1, List @@ e2][[1]];
  x = Complement[List @@ e1, {y}][[1]];
  z = Complement[List @@ e2, {y}][[1]];

  (* 创生新空间 *)
  maxNode = Max[VertexList[g]];
  w = maxNode + 1;

  (* 关键物理过程：相变 *)
  (* 新的活性表面 (Boundary) *)
  newActive = {UndirectedEdge[x, z], UndirectedEdge[x, w], UndirectedEdge[w, z]};
  (* 冻结的历史体积 (Bulk) *)
  newInert = {DirectedEdge[x, y], DirectedEdge[y, z]};

  (* 更新全图：移除旧活性边，加入新边 *)
  remainingEdges = Complement[allEdges, {e1, e2}];

  Graph[VertexList[g] ~Join~ {w}, remainingEdges ~Join~ newInert ~Join~ newActive]
];

(* 2. 全息扫描仪 (Holographic Scanner) *)
(* 从中心奇点向外扫描，测量每一层视界的面积与内部信息量 *)
ScanBlackHole[g_Graph] := Module[{
  simpleG, center, dists, maxR, scanData,
  nodesInside, subgraphInside,
  AreaA, InfoS, activeEdges
},
  (* 为了测量距离，我们将历史与当下视为统一的时空流形 *)
  simpleG = Graph[UndirectedEdge @@@ EdgeList[g]];

  (* 寻找中心奇点 (度数最大或初始点) *)
  center = First[SortBy[VertexList[simpleG], -VertexDegree[simpleG, #] &]];

  (* 计算时空度规 (距离场) *)
  dists = GraphDistance[simpleG, center] /. Infinity → 0;
  maxR = Max[dists];
```

```
  scanData = {};

  (* 逐层向外扫描 (从奇点到视界边缘) *)
  Do[
    (* 定义视界内部 (Bulk Volume) *)
    nodesInside = Pick[VertexList[g], UnitStep[r - dists], 1];

    (* S: 信息量 (Information) *)
    (* 定义为视界内部沉积的数量 (历史的重量) *)
    subgraphInside = Subgraph[g, nodesInside];
    InfoS = Count[EdgeList[subgraphInside], _DirectedEdge];

    (* A: 面积 (Area) *)
    (* 定义为穿过视界的数量 (正在发生的纠缠) *)
    (* 或者是视界的切割边数 (Cut Size) *)
    AreaA = 0;
    (* 统计连接 内→外 或 外→内 的边 *)
    (* 重点关注活性边，因为它们构成了全息屏 *)
    activeEdges = Cases[EdgeList[g], _UndirectedEdge];

    (* 计算穿过半径 r 的活性边数量 *)
    (* 注意：活性边通常在最外层，所以对于内部 r，AreaA 可能较小， *)
    (* 但在 AdS 中，表面积随半径指数增长，我们会看到 A 和 S 同步增长 *)
    AreaA = Count[activeEdges, UndirectedEdge[u_, v_] /;
      (MemberQ[nodesInside, u] || MemberQ[nodesInside, v])
    ];

    (* 过滤掉太小的数据点 *)
    If[AreaA > 0 && InfoS > 0,
      AppendTo[scanData, {AreaA, InfoS}]
    ];

  , {r, 1, maxR - 1}];

  scanData
];

(* 3. 运行实验 *)
```

```
RunHolographicTest[steps_] := Module[{universe, data, fit, plot},
 (* 初始种子：一个简单的闭环 *)
 universe = CycleGraph[4, DirectedEdges → False];

 Print[, steps, ];

 (* 演化 *)
 Monitor[
  Do[universe = StepEvolution[universe], {i, steps}],
  Row[{, i, , steps, , VertexCount[universe]}]
 ];

 Print[];
 data = ScanBlackHole[universe];

 (* 数据清洗：按面积排序 *)
 data = SortBy[data, First];

 If[Length[data] < 5, Print[]; Return[]];

 (* 线性拟合 S = k * A *)
 fit = LinearModelFit[data, x, x];

 Print[];
 Print[Style[ <> ToString[NumberForm[fit[[1, 1]], {4, 3}]], Red, Bold, 16]];

 (* 绘图 *)
 Show[
  ListPlot[data,
   PlotStyle → {Blue, PointSize[0.015]},
   AxesLabel → {, },
   PlotLabel → Style[, Bold, 14],
   GridLines → Automatic, Frame → True
  ],
  Plot[fit[x], {x, 0, Max[data[[All, 1]]]}, PlotStyle → {Red, Dashed}],
  Epilog → {

     Inset[Framed[Style[, Darker[Green], Bold], Background → White], Scaled[{0.2, 0.8}]]
  },
  ImageSize → 500
```

```
  ]
];
```

```
(* 建议运行 2000 步以上以获得清晰的 AdS 渐近行为 *)
RunHolographicTest[4000]
```

```
⬭ Generating AdS black hole (neuron bifurcation mode)... Evolution4000 steps
Evolution completed. Holistic tomography scan in progress....

📐 Holographic coefficient determination(Holographic Coefficient):
slope
   =

                -8
{{, Estimate, Standard Error, t-Statistic,
  P-Value}, {1, -623.600, 351.500, -1.774,
  0.110}, {x, 2.414, 0.155, 15.570, 8.179 × 10   }}
```

Out[47]=