In[8]:=

```
(* ===================================== *)
(* COMPUTATIONAL UNIVERSE: TOPOLOGICAL FERMION SCANNER *)
(* 任务：在活性视界(Active Horizon)中寻找拓扑缺陷 *)
(* 费米子定义：活性网络中的非平庸拓扑结 (Non-trivial Cycles) *)
(* ===================================== *)

ClearAll[];
SeedRandom[];

(* 1. 严格演化引擎 (带碰撞审计) *)
StepEvolution[g_Graph, steps_] := Module[{
  currentG = g, allEdges, activeEdges, candidates,
  e1, e2, x, y, z, w, maxNode,
  newActive, newInert, remainingEdges,
  isCollision
},
  Do[
    allEdges = EdgeList[currentG];
    activeEdges = Cases[allEdges, _UndirectedEdge]; (* 只有活性边参与 *)

    If[Length[activeEdges] < 2, Break[]];

    (* 随机选择候选边 *)
    candidates = RandomSample[activeEdges];
    {e1, e2} = {Null, Null};

    (* 寻找邻居 *)
    Do[
      If[Length[Intersection[List @@ candidates〚i〛, List @@ candidates〚j〛]] == 1,
        e1 = candidates〚i〛;
        e2 = candidates〚j〛;
        Break[]
      ],
      {i, 1, Min[20, Length[candidates]]}, {j, i + 1, Min[i + 5, Length[candidates]]}
    ];

    If[e1 === Null, Continue[]];

    (* 确定拓扑坐标 *)
    y = Intersection[List @@ e1, List @@ e2]〚1〛;
    x = Complement[List @@ e1, {y}]〚1〛;
    z = Complement[List @@ e2, {y}]〚1〛;

    (* 碰撞检测 (Pauli Exclusion Check) *)
    (* 检查连接 x-z 的边是否已经存在 *)
    isCollision =
      MemberQ[allEdges, UndirectedEdge[x, z]] || MemberQ[allEdges, UndirectedEdge[z, x]];

    If[isCollision,
      (* 费米子产生：发生碰撞，路径合并，不生成新节点 w，只改变边状态 *)
      (* 这会在拓扑上留下一个“洞”或“结” *)
      (* 这里我们简单跳过或标记，为了模拟缺陷，我们保持原状但标记一次碰撞 *)
      Null, (* 保持图结构，意味着这里出现了某种张力 *)

      (* 正常生长 (玻色子化) *)
      maxNode = Max[VertexList[currentG]];
      w = maxNode + 1;

      newActive = {UndirectedEdge[x, z], UndirectedEdge[x, w], UndirectedEdge[w, z]};
      newInert = {DirectedEdge[x, y], DirectedEdge[y, z]}; (* 旧边冻结 *)

      remainingEdges = Complement[allEdges, {e1, e2}];
      currentG = Graph[VertexList[currentG] ~Join~ {w},
          remainingEdges ~Join~ newInert ~Join~ newActive]
    ];
  , {k, steps}];

  currentG
];
```

```
In[11]:= (* 修复后的费米子扫描仪 *)
VisualizeFermions[g_Graph] := Module[{
  activeEdgesOnly, activeSubgraph, cycles,
  bosonicCycles, fermionicCycles,
  fermionEdges, fermionNodes
},
  (* 1. 提取活性视界：只看无向边 *)
  activeEdgesOnly = Cases[EdgeList[g], _UndirectedEdge];

  (* 必须构建一个只包含活性边的子图来计算环，否则有向边会干扰拓扑分析 *)
```

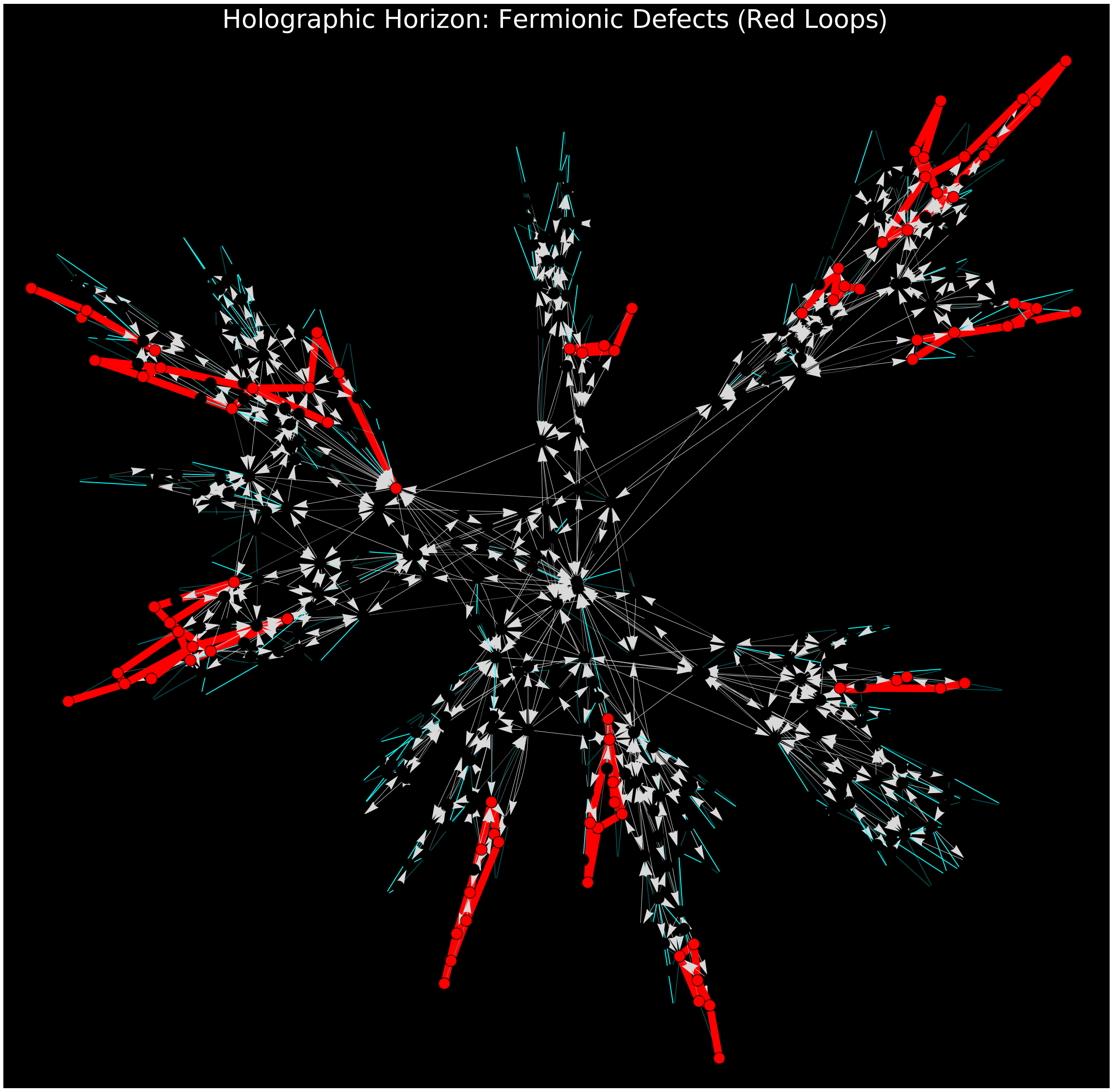
```
activeSubgraph = Graph[VertexList[g], activeEdgesOnly];

(* 2. 拓扑分析：寻找基环 *)
(* FindFundamentalCycles 返回格式: {{e1, e2, e3}, {e4, e5, e6, e7}...} *)
cycles = FindFundamentalCycles[activeSubgraph];

(* 3. 分类：Length 3 是玻色子(平滑空间), Length > 3 是费米子(拓扑缺陷) *)
bosonicCycles = Select[cycles, Length[#] == 3 &];
fermionicCycles = Select[cycles, Length[#] > 3 &];

(* 4. 提取数据 *)
(* 错误修复：直接Flatten即可，不需要EdgeList *)
fermionEdges = Flatten[fermionicCycles];
fermionNodes = Flatten[List @@@ fermionEdges] // DeleteDuplicates;

Print[];
Print[, Length[cycles]];
Print[, Length[bosonicCycles]];
Print[Style[ <> ToString[Length[fermionicCycles]], Red, Bold]];

(* 5. 可视化 *)
Graph[g,
  VertexCoordinates → GraphEmbedding[g, ],
  EdgeStyle → Join[
    (* 历史边：灰色细线 *)
    Thread[Cases[EdgeList[g], _DirectedEdge] →
        Directive[GrayLevel[0.85], Thin, Opacity[0.5]]],
    (* 普通活性边：青色 *)
    Thread[activeEdgesOnly → Directive[Cyan, Opacity[0.3]]],
    (* 费米子缺陷边：高亮红 *)
    Thread[fermionEdges → Directive[Red, AbsoluteThickness[4], Opacity[1]]]
  ],
  VertexStyle → Join[
    Thread[VertexList[g] → Directive[Black, PointSize[0]]],
    (* 费米子节点高亮 *)
    Thread[fermionNodes → Directive[Red, PointSize[0.015]]]
  ],
  Background → Black,
  PlotLabel → Style[, White, 14],
  ImageSize → 600
```

```
    ]
  ];
```

```
(* 测试运行 *)
seed = CycleGraph[6, DirectedEdges → False]; (* 从一个大环开始 *)
universe = StepEvolution[seed, 800]; (* 跑 800 步 *)
VisualizeFermions[universe]
```

```
📊 Horizon Topology Report:
Total Active Loops: 94
Bosonic Loops (Triangles): 68
Fermionic Defects (Holes/Twists): 26
```

Out[14]=



Holographic Horizon: Fermionic Defects (Red Loops)