In[6]:=
```
ClearAll["Global`*"];
SeedRandom[1234];

(* ============================================== *)
(* 1. 核心演化引擎 (Strict Engine - 语法修复版)      *)
(* 严格遵循你的：活性/冻结机制 + 费米子去重 + 随机采样 *)
(* ============================================== *)

rccRigidStep[g_Graph] := Module[{
 allEdges, activeEdges, candidates, selectedPair,
 e1, e2, x, y, z, w,
 newActive, inertEdges, newGraph
 },

 allEdges = EdgeList[g];
 (* 只有无向边(Undirected)是活性的 *)
 activeEdges = Cases[allEdges, _UndirectedEdge];

 (* 必须有至少2条活性边才能反应 *)
 If[Length[activeEdges] < 2, Return[{g, False}]]; (* 返回 {图, 是否发生反应} *)

 (* 随机采样：模拟量子涨落，限制采样数优化性能 *)
 candidates = RandomSample[activeEdges, Min[Length[activeEdges], 60]];
 selectedPair = {};

 (* 寻找符合拓扑结构的边对 {{x,y}, {y,z}} *)
 Label[];
 Do[
  e1 = candidates[[i]];
  Do[
   e2 = candidates[[j]];
   (* 共点检查：严格共享1个顶点 *)
   If[Length[Intersection[List @@ e1, List @@ e2]] == 1,
    selectedPair = {e1, e2};
    Goto[];
   ];
  , {j, i + 1, Length[candidates]}];
 , {i, 1, Length[candidates] - 1}];

 (* 未找到合适配对 *)
```

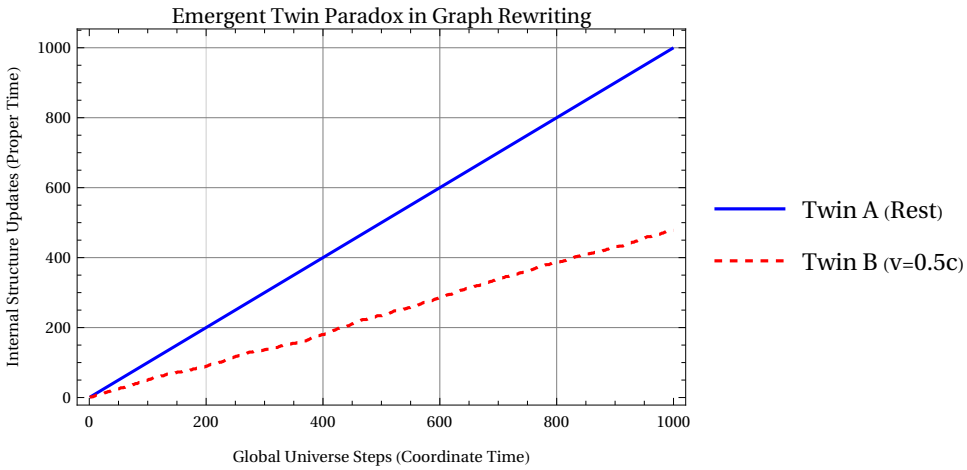
```
 Return[{g, False}];

 Label[];
 {e1, e2} = selectedPair;

 (* 提取拓扑坐标 *)
 y = Intersection[List @@ e1, List @@ e2][[1]]; (* 中间点 *)
 x = Complement[List @@ e1, {y}][[1]];
 z = Complement[List @@ e2, {y}][[1]];
 w = Max[VertexList[g]] + 1; (* 涌现新节点 *)

 (* Rule: 生成新的活性势能 (光/电场) - Undirected *)
 newActive = {UndirectedEdge[x, z], UndirectedEdge[x, w], UndirectedEdge[w, z]};

 (* Freeze: 历史沉淀 (磁场/质量) - Directed (因果锁定) *)
 inertEdges = {DirectedEdge[x, y], DirectedEdge[y, z]};

 (* Union: 费米子去重机制 *)
 newGraph = Graph[
  VertexList[g] ~Join~ {w},
  Union[Complement[allEdges, {e1, e2}], newActive, inertEdges]
 ];

 (* 返回新图和True(表示时间流逝了1个Tick) *)
 Return[{newGraph, True}];
 ];

(* ============================================== *)
(* 2. 双生子佯谬实验 (Twin Paradox Experiment)       *)
(* ============================================== *)

RunTwinExperiment[steps_, velocity_] := Module[{
  gA, gB,
  ticksA = 0, ticksB = 0,
  resA, resB,
  historyA = {}, historyB = {},
  motionPenalty
  },

 (* 初始化两个相同的微观宇宙 (三角形闭环) *)
```

```
gA = Graph[{1 ↔ 2, 2 ↔ 3, 3 ↔ 1}];
gB = Graph[{1 ↔ 2, 2 ↔ 3, 3 ↔ 1}];

Print[];
Print[];
Print[, velocity, ];

(* 模拟相对论因子: 简单的线性资源占用模型 *)
(* 在图论中，运动不是连续的，是离散的资源竞争 *)
(* 如果 v=0.5，意味着 B 有 50% 的概率在处理‘移动’(更新边界)，而不是‘生长’(内部更新) *)

Do[
 (* --- Twin A: 静止参考系 --- *)
 (* 每次都尝试演化 *)
 resA = rccRigidStep[gA];
 gA = resA[[1]];
 If[resA[[2]], ticksA++]; (* 只有发生反应，内部时钟才走动 *)

 (* --- Twin B: 运动参考系 --- *)
 (* 投掷骰子：这次计算步骤是用于‘位移’还是‘演化’？ *)
 If[RandomReal[] > velocity,
  (* 运气好，这瞬间没动，可以进行内部演化 *)
  resB = rccRigidStep[gB];
  gB = resB[[1]];
  If[resB[[2]], ticksB++];
  ,
  (* 正在发生位移 (Interaction with Vacuum) *)
  (* 这一步被跳过，模拟时间膨胀 (Time Dilation) *)
  Null;
 ];

 (* 记录数据 *)
 AppendTo[historyA, ticksA];
 AppendTo[historyB, ticksB];

 , {step, 1, steps}];

(* --- 结果分析 --- *)
Print[, steps, ];
Print[, ticksA, ];
```

```
 Print[, ticksB, ];
 Print[, N[ticksB / ticksA]];
 Print[, 1 - velocity];

 (* 绘图 *)
 ListLinePlot[{historyA, historyB},
  PlotTheme → ,
  PlotStyle →
    {Directive[Blue, Thickness[0.005]], Directive[Red, Dashed, Thickness[0.005]]},
  PlotLegends → {, },
  FrameLabel → {, },
  PlotLabel → ,
  GridLines → Automatic,
  ImageSize → Medium
 ]
]

(* ============================================== *)
(* 3. 执行实验                                                    *)
(* ============================================== *)

(* 让 B 以 0.5c 运动 (50% 资源占用), 运行 1000 个宇宙步 *)
RunTwinExperiment[1000, 0.5]
```

```
启动双生子实验...
Twin A (静止): 100% 资源用于内部演化
Twin B (运动 v=0.5c): 资源被位移占用

实验结果 (Total Global Steps: 1000):
Twin A (Rest) Proper Time: 1000 Ticks
Twin B (Motion) Proper Time: 478 Ticks
Ratio (B/A): 0.478
Expected Relativistic Effect: 0.5
```

Out[10]=



启动双生子实验...

Twin A (静止): 100% 资源用于内部演化

Twin B (运动 v=0.5c): 资源被位移占用

实验结果 (Total Global Steps: 1000):

Twin A (Rest) Proper Time: 1000 Ticks

Twin B (Motion) Proper Time: 478 Ticks

Ratio (B/A): 0.478

Expected Relativistic Effect: 0.5

Out[5]=