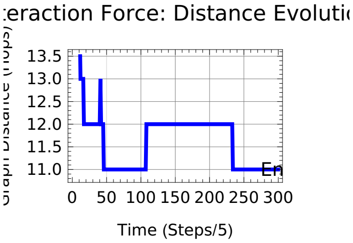
```
In[57]:= ClearAll["Global`*"];
         (* ================================================================ *)
         (* 1. 核心引擎：严格遵循 Rule + 冻结机制 *)
         (* ================================================================ *)
         rccRigidStep[g_Graph] :=
          Module[{allEdges, activeEdges, selectedPair, e1, e2, x, y, z, w,
            nextV, newActiveEdges, inertEdges, candidates},
           allEdges = EdgeList[g];
           (* 核心公理：只有无向边(当下)参与反应，有向边(历史)不参与 *)
           activeEdges = Cases[allEdges, _UndirectedEdge];
           If[Length[activeEdges] < 2, Return[g]];

           (* 寻找反应对：模拟底层量子涨落的随机采样 *)
           candidates = {};
           Block[{shuffled = RandomSample[activeEdges, Min[Length[activeEdges], 80]]},
            Do[
             e1 = shuffled[[i]];
             (* 寻找共享中心点 y 的邻边 *)
             Do[
              e2 = shuffled[[j]];
              If[Length[Intersection[List @@ e1, List @@ e2]] == 1,
               selectedPair = {e1, e2};
               Goto[]
               ], {j, i + 1, Length[shuffled]}
              ], {i, 1, Length[shuffled]}
             ]
            ];
           Return[g]; (* 未找到可反应对 *)

           Label[];
           {e1, e2} = selectedPair;
           y = Intersection[List @@ e1, List @@ e2][[1]];
           x = Complement[List @@ e1, {y}][[1]];
           z = Complement[List @@ e2, {y}][[1]];

           nextV = Max[VertexList[g]] + 1;
           w = nextV;

           (* The Rule: 生成新的活性空间 *)
           newActiveEdges = {UndirectedEdge[x, z], UndirectedEdge[x, w],
    UndirectedEdge[w, z]};

   (* Causal Fixation: 旧路径冻结为历史 (有向边) *)
   (* 这种转化导致了粒子性的 *)
   inertEdges = {DirectedEdge[x, y], DirectedEdge[y, z]};

   Graph[VertexList[g]~Join~{w},
    Union[Complement[allEdges, {e1, e2}], newActiveEdges, inertEdges]]
   ];

(* ============================================================== *)
(* 2. 手性粒子构建器 (Topological Chirality Constructor) *)
(* ============================================================== *)
(* 构建一个具有特定缠绕方向的高密度团块 *)
CreateChiralParticle[startIdx_, size_, chirality_String] :=
 Module[{nodes, edges, twists},
  nodes = Range[startIdx, startIdx + size - 1];
  (* 基础骨架：高密度环 *)
  edges = UndirectedEdge @@@ Partition[nodes, 2, 1, 1];

  (* 注入手性：通过不对称的内部连线定义或 *)
  (* 这里的逻辑是：左旋粒子倾向于连接 (i, i+2)，右旋连接 (i, i-2) 模拟拓扑扭曲 *)
  twists = Switch[chirality,
    ,
    Table[UndirectedEdge[nodes[[i]], nodes[[Mod[i + 2 - 1, size] + 1]]], {i, 1, size}],
    ,
    Table[UndirectedEdge[nodes[[i]], nodes[[Mod[i - 2 - 1, size] + 1]]], {i, 1, size}],
    , {}
    ];

  {nodes, Union[edges, twists]}
  ];

(* ============================================================== *)
(* 3. 实验设置：真空中的双体互动 *)
(* ============================================================== *)
(* 实验参数：你可以修改这里来测试 + (异性) 或 + (同性) *)
particle1Type = ;
particle2Type = "Left"; (* 试着改成 "Left" 看看排斥效应 *)
```

```
Print[Style["loading: " <> particle1Type <> " vs " <> particle2Type, Blue, Bold]];

(* 1. 构建真空背景 (稀疏网格) *)
vacuumWidth = 8;
vacuumLength = 10;
gVacuum = GridGraph[{vacuumLength, vacuumWidth}];
vVacuum = VertexList[gVacuum];
eVacuum = EdgeList[gVacuum];

(* 2. 注入粒子 A (左侧) *)
{nodesA, edgesA} = CreateChiralParticle[Max[vVacuum] + 1, 6, particle1Type];
(* 将粒子 A 挂载到真空左端 *)
linkA = {UndirectedEdge[nodesA⟦1⟧, 1], UndirectedEdge[nodesA⟦2⟧, 2]};

(* 3. 注入粒子 B (右侧) *)
{nodesB, edgesB} = CreateChiralParticle[Max[nodesA] + 1, 6, particle2Type];
(* 将粒子 B 挂载到真空右端 *)
linkB = {UndirectedEdge[nodesB⟦1⟧, vacuumLength*vacuumWidth],
  UndirectedEdge[nodesB⟦2⟧, vacuumLength*vacuumWidth - 1]};

(* 4. 组装初始宇宙 *)
initG = Graph[
  Join[vVacuum, nodesA, nodesB],
  Join[eVacuum, edgesA, edgesB, linkA, linkB],
  GraphLayout →
  ];

(* ============================================================== *)
(* 4. 运行演化与测量 *)
(* ============================================================== *)
steps = 1500; (* 步数 *)
Print[Style["Start the causal evolution engine...", Red]];

(* 监控数据：两个粒子核心之间的图距离 *)
distanceData = {};
interactionData = {}; (* 记录中间真空区的活性 *)

monitorG = initG;
CheckDistance[g_] :=
 Module[{dist},
  (* 测量粒子A核心与粒子B核心的最短路径 *)
  dist = GraphDistance[g, nodesA〚1〛, nodesB〚1〛];
  If[IntegerQ[dist], dist, 100] (* 如果断开则设为最大值 *)
  ];

Monitor[
 Do[
  monitorG = rccRigidStep[monitorG];

  (* 每5步采样一次数据，避免拖慢速度 *)
  If[Mod[i, 5] == 0,
   AppendTo[distanceData, CheckDistance[monitorG]];
   (* 简单的真空极化度量：总边数变化 *)
   AppendTo[interactionData, EdgeCount[monitorG]];
   ];
  , {i, 1, steps}],
 Row[
   {ProgressIndicator[i, {0, steps}], " Step: ", i, " | Distance: ", Last[distanceData]}]
 ];

Print[Style[
   "☑ The experiment is complete! The analysis report is being generated...", Green]];

(* ================================================================= *)
(* 5. 结果可视化：引力/库仑力分析 *)
(* ================================================================= *)
finalG = monitorG;

(* 图1：距离演化 (验证吸引/排斥) *)
plotDist = ListLinePlot[distanceData,
  Frame → True, FrameLabel → {, },
  PlotLabel → ,
  PlotStyle → Directive[Thick, Blue],
  GridLines → Automatic,
  Epilog → {
   Text[Style[, 10], {5, distanceData〚1〛}],
   Text[Style[, 10], {Length[distanceData], Last[distanceData]}]
   }];
```

```
(* 图2：最终拓扑结构 (可视化连接) *)
(* 仅高亮活性边，看看是否形成了连接通道 *)
vizFinal = Graph[VertexList[finalG], EdgeList[finalG],
 VertexSize → 0,
 EdgeStyle → {
  _DirectedEdge → Directive[Opacity[0.1], Gray], (* 历史背景 *)
  _UndirectedEdge → Directive[Opacity[0.8], Orange] (* 活性连接 *)
 },
 GraphLayout → ,
 PlotLabel → ,
 ImageSize → 400];

Grid[{{plotDist}, {vizFinal}}]
```

```
loading: Left vs Left
Start the causal evolution engine...
☑ The experiment is complete! The analysis report is being generated...
```



Final Topological State (Orange = Active Force Fields)