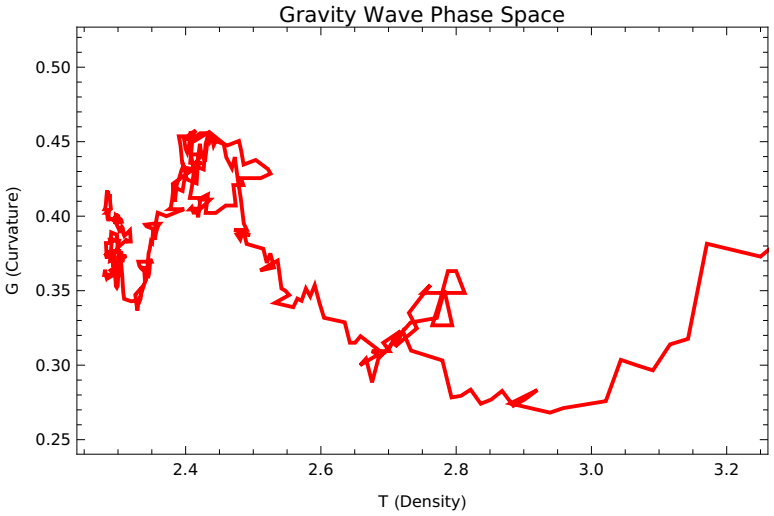
In[1]:=
```
ClearAll["Global`*"];
SeedRandom[777]; (*幸运种子*)
(* ============================================================*)
(*1. 核心规则引擎 (修复了删除逻辑)*)
(* ============================================================*)
rccRigidStep[g_Graph] :=
  Module[{allEdges, activeEdges, candidates, selectedPair, e1, e2, x,
    y, z, w, nextV, newActiveEdges, inertEdges, found},
   allEdges = EdgeList[g];
   activeEdges = Cases[allEdges, _UndirectedEdge];
   If[Length[activeEdges] < 2, Return[g]];
   (*---寻找反应对---*)found = False;
   Block[{shuffled =
      RandomSample[activeEdges, Min[Length[activeEdges], 50]]},
    Do[e1 = shuffled[[i]];
     (*寻找有一点重合的边*)
     candidates =
      Select[shuffled, (Length[Intersection[List @@ e1, List @@ #]] ==
           1) &];
     If[Length[candidates] > 0, e2 = First[candidates];
      selectedPair = {e1, e2};
      found = True;
      Break[];], {i, 1, Length[shuffled]}]];
   If[Not[found], Return[g]];
   (*---执行重写---*){e1, e2} = selectedPair;
   y = Intersection[List @@ e1, List @@ e2][[1]]; (*中心点*)
   x = Complement[List @@ e1, {y}][[1]];
   z = Complement[List @@ e2, {y}][[1]];
   nextV = Max[VertexList[g]] + 1;
   w = nextV;
   (*新生：活性波前*)
   newActiveEdges = {UndirectedEdge[x, z], UndirectedEdge[x, w],
     UndirectedEdge[w, z]};
   (*冻结：历史路径*)
   inertEdges = {DirectedEdge[x, y],
     DirectedEdge[y,
      z]};
   (*严格遵循流向*)(*---关键修复：直接从列表中移除 e1 和 e2 对象---*)(*之前的方法如果边的方向不同会删除失败，现在直接移除选中的对象*)
```

```
    Graph[VertexList[g]~Join~{w},
     Union[Complement[allEdges, {e1, e2}], (* <---这里修复了 Bug*)
      newActiveEdges, inertEdges]]];
(* ================================================================*)
(*2. 初始条件：双体引力场*)
(* ================================================================*)
Print[Style[, Blue]];
(*团块 A*)
vA = Range[1, 8];
eA = EdgeList[CompleteGraph[8]];
(*团块 B (偏移索引)*)
vB = Range[9, 14];
(*安全的偏移方法*)
eB = EdgeList[CompleteGraph[6]] /. (i_Integer :> i + 8);
(*活性桥*)
bridge = {UndirectedEdge[RandomChoice[vA], RandomChoice[vB]]};
initG = Graph[Join[vA, vB], Join[eA, eB, bridge],
   GraphLayout → "SpringElectricalEmbedding"];
(* ================================================================*)
(*3. 运行演化 (带进度条)*)
(* ================================================================*)
steps = 400;
Print[Style["start...", Red]];
(*使用 Monitor 显示进度，防止您以为卡死了*)
evolutionHistory = {initG};
Monitor[Do[
   AppendTo[evolutionHistory,
    rccRigidStep[Last[evolutionHistory]]], {i, 1, steps}],
  Row[{ProgressIndicator[i, {0, steps}], " Step: ", i, "/", steps}]];
finalG = Last[evolutionHistory];
Print[Style["☑ finished！", Green]];
(* ================================================================*)
(*4. 可视化 (确保出图)*)
(* ================================================================*)
Print[Style[, Purple]];
AnalyzeMetric[g_] :=
  Module[{activeSubgraph, valG},
   activeSubgraph = Graph[Cases[EdgeList[g], _UndirectedEdge]];
   If[EdgeCount[activeSubgraph] == 0, Return[{0, 0}]];
```

```
    {N[Mean[VertexDegree[activeSubgraph]]], (*T:密度*)
     GlobalClusteringCoefficient[activeSubgraph] (*G:曲率 (稳健)*)}];
(*提取数据*)
waveData = AnalyzeMetric /@ Drop[evolutionHistory, 10];
(*绘图*)
Row[{(*左图：相空间*)
  ListLinePlot[waveData, Frame → True,
   FrameLabel → {, },
   PlotLabel → , PlotStyle → Red,
   ImageSize → 400], (*右图：3D 结构*)
  Graph3D[finalG, GraphLayout → ,
   VertexSize → 0,
   EdgeStyle → {_UndirectedEdge →
      Directive[Opacity[1], White, Thickness[0.005]], _DirectedEdge →
       Directive[Opacity[0.2], Blue, Thin]}, Boxed → False,
   Background → Black, ImageSize → 400,
   PlotLabel → ]}]
loading...
start...
☑ finished!
loading...
```

Out[20]=


Gravity Wave Phase Space
G (Curvature)
0.50
0.45
0.40
0.35
0.30
0.25
2.4
2.6
2.8
3.0
3.2
T (Density)