面向 3D 芯片网表切分的优化目标演化

“韬”智能杯 EDA 创新大赛--基于自演化智能体的 3D 芯片物理设计算法优化[1]

一、赛题背景

集成电路长期依靠晶体管尺寸缩小来提高集成密度，并在单位晶体管成本、性能和能效方面持续受益。随着制造工艺不断接近物理与工程极限，进一步缩小器件面临更高的制造成本和漏电流控制难度，供电电压难以继续下降也使芯片功率密度不断上升。与此同时，芯片内部的数据传输距离和互连延迟逐渐成为性能提升的重要制约。如何在有限工艺条件下继续提高芯片集成能力、延续摩尔定律所代表的发展趋势，已成为先进芯片设计的重要问题。

三维芯片将电路从二维平面扩展到垂直堆叠的多个有源层，通过混合键合等高密度层间互连技术把多个有源层连接起来，使信号可在垂直方向传输，从而在有限投影面积内集成更多逻辑与存储资源，并缩短部分长距离连接。逻辑折叠（Logic Folding）在此基础上进一步细化划分粒度：它不局限于沿既有模块边界切分，而是将同一逻辑块内部原本铺展在一个平面上的单元分布到不同层，使关键路径上较长的水平连接转化为较短的垂直连接。这为突破单纯依赖工艺微缩的限制提供了新的途径，但收益仍取决于电路如何划分到多个晶片层、单元如何放置，以及后续设计环节能否保留这些收益[1–2]。

网表切分决定每个单元属于哪一晶片层，是三维物理设计的重要起点。切分目标既要考虑面积分配，也要优化线长与时序需求。Open3DBench 为三维芯片后端实现与评价提供了开放基础[3]。本赛题在此基础上引入基于大模型的算法演化，由智能体读取代码和评估反馈，迭代改进切分算法的目标函数，研究如何提升切分优化目标与实际物理收益的一致性。

二、赛题应用场景

先进移动芯片、通用处理器、人工智能加速器及高带宽计算系统，都面临更高集成密度、更低通信开销和更好能效的需求。三维集成既可用于逻辑与存储堆叠，也可用于逻辑与逻辑堆叠。面向后者的逻辑折叠技术可以将联系紧密的单元分置于上下相邻的层，从而缩短部分关键路径，并为同等性能下的降压和节能创造条件。

华为提出的“韬（τ）定律”以缩短信号传播的特征时间为核心，将优化视角从器件几何尺寸扩展到电路、芯片和系统[1]。其公开论文以麒麟片上系统（System on

1 EDA 为 Electronic Design Automation（电子设计自动化）。

Chip，SoC）为代表，介绍了逻辑折叠在处理器和计算模块中的应用[2]。这类设计对面积、性能和功耗的要求，对网表切分与后续布局等流程的算法设计提出了新的挑战。本赛题使用开源基准芯片研究双层逻辑设计中的网表切分方法。

三、赛题任务

参赛者需实现智能体运行框架（Agent Harness，下称 harness），使其能够调用统一基座模型，读取初始算法、生成和修改代码、提交候选评估，并根据日志与指标选择后续方案。交付成果包括选定的最终 harness、由其生成并选定的算法代码，以及对应的切分和布局结果。

网表描述逻辑单元、宏块及其连接关系，可表示为超图。以标准单元或宏块为节点，以连接多个引脚的线网为超边。网表切分通过确定各节点的上下层归属，将网表划分至不同芯片层，从而影响各层的面积占用、跨层连接数量及后续布局条件。其中，跨层线网通过混合键合端子（Hybrid Bonding Terminal，HBT）实现连接。

传统切分方法主要关注割网数量最小化与层间面积均衡，较少考虑切分结果对后续线长、时序等物理设计指标的影响。近年来，相关研究利用图神经网络（Graph Neural Network，GNN）学习网表的连接结构与节点时序特征，形成节点的深层特征表示，并通过表示空间中的聚类生成离散切分方案[4]。这类方法通过引入时序相关特征，在一定程度上增强了切分过程的时序感知能力。然而，其时序信息的利用仍主要局限于特征表征层面，其 GNN 的自监督学习目标侧重于挖掘图结构信息，缺乏对下游时序指标的直接优化。

针对上述局限，本赛题引入智能体自演化方法，以后续物理设计流程中的时序指标作为适应度反馈，迭代演化 GNN 切分过程中的训练损失函数与聚类机制，探索与下游时序优化目标更加一致的切分目标建模方法。赛题提供两种基于 GNN 的切分基线：第一种参考文献[4]，采用表示学习与表示空间聚类相结合的方式生成切分方案；第二种采用 GNN 直接预测切分结果，将割网数量与面积均衡目标构建为可微损失函数，实现端到端的无监督训练。参赛队伍可任选一种基线，利用智能体自演化方法优化相应的损失函数设计及适用的聚类机制，最终基于统一的物理设计指标进行评测与排名。

（一）表示学习与聚类切分基线

该基线采用图神经网络学习节点的深层特征表示，再通过聚类方法生成上下层切分方案。初始方法采用基于正负样本对比的无监督损失函数，以邻近节点作为正样本、随机采样节点作为负样本，通过拉近正样本对的特征表示、拉远负样本对的

特征表示，学习网表的连接结构特征。训练完成后，采用面积加权的两路聚类方法，将连续节点表示转化为离散切分结果。

该基线开放训练损失函数与聚类解码函数作为演化目标，参赛队伍可选择单独优化或联合演化。损失函数的演化旨在改进节点表示的学习目标，更充分地利用连接结构、面积及时序信息；聚类机制的演化则旨在优化离散分组策略，使切分结果更好地兼顾面积均衡与下游时序质量。参赛方案须保持规定的函数接口。

（二）概率预测切分基线

图神经网络直接预测各节点的上下层归属概率，通过固定阈值生成初始切分方案，并由官方程序完成约束修复。初始损失函数综合考虑割网数量、面积均衡及预测概率的二值化约束，实现端到端的无监督训练。

该基线开放损失函数作为演化目标。由于该基线下损失函数与优化目标直接相关，且不必遵循固定的自监督学习形式，所以智能体可以更自由地探索有利于下游时序质量的函数建模。参赛方案须保持规定的函数接口。

（三）共同物理评估

两种基线生成的切分结果均进入统一的三维芯片物理设计流程，依次完成布局与合法化、供电网络构建、跨层端子分配、时序优化及静态时序分析。布局采用基于 DREAMPlace 的实现[5]；用于计分的静态时序分析（Static Timing Analysis，STA）在时序优化之后、时钟树综合之前执行，在连通的双层时序视图中采用布局估算的互连寄生参数、理想时钟及固定的无源 HBT 电气模型。布局及后续环节的源码与参数不属于本赛题的演化范围。相关物理设计环节可参阅 OpenROAD 官方文档[6]。

四、数据集及数据说明

（一）数据来源与划分

赛题采用五个公开算例和两个隐藏测试算例，提供网表、工艺库、时序约束、图及节点特征。工艺设计套件（Process Design Kit，PDK）提供器件、布线层和设计规则。算例名中的 ng45 对应 Nangate 45 nm 开放库平台，asap7 对应亚利桑那州立大学 7 nm 预测工艺平台（ASAP7）。两种基线使用相同数据。

训练集包含 bp_fe-ng45、bp-ng45、ariane133-ng45 三个算例，覆盖不同网表结构和规模，用于 harness 的候选生成、反馈分析和算法选择。

公开验证集包含 swerv_wrapper-ng45、bp_fe-asap7，用于候选确定后的泛化检验。正式演化运行中，验证集结果不用于继续修改或选择候选。bp_fe-asap7 与训

练集的 bp_fe-ng45 电路来源相同、工艺条件不同，可检验跨工艺适应能力。最终评测会加入两个隐藏测试算例，七个算例均计入最终成绩。

（二）随机种子

每个算例使用 1001、1002、1003 三个随机种子，以减小算法的随机性影响。每个种子均与该算例的官方共同评分参考值比较，先计算各自得分，再对同一算例的三个种子取算术平均。

（三）输入文件与格式

1. 电路网表（.v）。采用 Verilog 文本格式，描述模块端口、标准单元实例及引脚之间的连接。网表说明“哪些单元相连”，通常不描述其物理坐标。以下是简化示例，实际单元类型及引脚名称应与提供的库一致。

```
module example(input a, input b, output y);
  wire n1;
  NAND2_X1 U1 (.A1(a), .A2(b), .ZN(n1));
  INV_X1   U2 (.A(n1), .ZN(y));
endmodule
```

2. 物理设计文件（Design Exchange Format，DEF，.def）。描述设计边界、坐标单位、放置行、外部引脚、单元位置及网络等物理信息。输入 DEF 可提供布局规划，输出 DEF 记录算法产生的放置结果。下例为片段：每微米对应 1000 个数据库单位，U1 的位置为（10 μm，20 μm），朝向 N 表示标准朝向。

```
VERSION 5.8 ;
DESIGN example ;
UNITS DISTANCE MICRONS 1000 ;
DIEAREA ( 0 0 ) ( 100000 100000 ) ;
COMPONENTS 2 ;
- U1 NAND2_X1 + PLACED ( 10000 20000 ) N ;
- U2 INV_X1   + PLACED ( 12000 20000 ) N ;
END COMPONENTS
# Rows, pins and nets are omitted in this excerpt.
```

3. 库与约束。库交换格式（Library Exchange Format，LEF，.lef）描述布线层、过孔规则，以及单元尺寸、引脚和遮挡区域；Liberty 库（.lib）提供单元时序、功

耗与负载模型。Synopsys 设计约束（Synopsys Design Constraints，SDC，.sdc）定义时钟、输入输出延迟等分析条件，例如“create_clock -name core_clk -period 1.0 [get_ports clk]”定义周期为 1 ns 的时钟。

4. 二维情况下的时序特征。时序特征来自于同算例的二维芯片物理设计的时序分析结果，能够为三维芯片的优化提供参考。每个节点的具体特征包括时序裕量、转换时间、到达与要求时间、单元延迟、时序覆盖比例和图连接信息等等。

五、算法设计要求

（一）演化对象和算法约束

概率预测基线仅开放损失函数，表示学习与聚类基线开放损失函数和聚类解码函数。图构建与收缩、节点特征、图神经网络结构、表示归一化、优化器、采样、学习率和训练预算等其他环节均保持固定。候选函数须遵循任务包已发布的数值接口规范，具体函数签名及允许的操作见 README 和对应任务的接口说明。

合法的切分结果应保持原电路逻辑功能和网表连接关系，每个原始实例须且仅须分配至一个芯片层。层间面积平衡要求为：上层实例总面积占两层实例总面积的 47%—53%；存在宏块时，上层宏块数量占比和宏块面积占比均须为 45%—55%。切分经修复程序后必须满足上述约束。

（二）Harness 设计

Harness 应组织“提出修改—生成代码—检查代码—提交评估—分析反馈—选择方案”的算法演化循环，并保存可追溯的算法候选历史。参赛者可设计和改进提示词、搜索策略、失败诊断和候选保留等环节。

Harness 可由参赛者人工设计，也可由智能体修改 harness 自身的代码、提示模板和执行逻辑，迭代提升其分析反馈与修改算法的能力，实现算法修改能力的自演化。参赛者自行选定最终提交的最优 harness 版本，赛题不限定该版本的形成方式，离线专家知识的使用须遵守第十一章规定。算法演化与 harness 自我改进的相关研究可参见文献[7–8]。

本赛题统一使用 DeepSeek-V4.1-Flash 作为基座模型，通过应用程序编程接口（Application Programming Interface，API）调用。该模型负责生成和修改代码，与切分算法中的图神经网络作用不同。官方以最终提交的 harness 为评测对象，从所选基线的官方初始算法出发，在相同基模和预算下运行，以所得算法的物理表现衡量其迭代改进算法的能力。提交时须固定最终 harness、提示模板、配置和基线选择，供组委会统一复跑。

六、性能指标要求

（一）时序指标与改善比例

时序指标采用最差负时序裕量（Worst Negative Slack，WNS）和总负时序裕量（Total Negative Slack，TNS），单位为纳秒。前者描述最严重的建立时间违例，后者汇总所有受约束端点的负建立时间裕量。对每项指标，以 $\max(0, -\text{指标值})$ 定义非负违例量。该违例量越小，时序越好。

单项改善比例＝（该算例的共同参考违例量－候选违例量）÷该算例的共同参考违例量

每个算例分别设置固定的 WNS 与 TNS 共同参考违例量，且两项参考值均为正。同一算例的两条 GNN 基线及三个种子共用这组参考值，WNS 与 TNS 分别计算改善比例。改善比例为正表示违例减少，为负表示退化。共同参考值由官方依据两条基线的参考测量统一校准并固定，五个公开算例的参考表使两条基线在应用得分下限之前的等权平均原始分相同。所有队伍使用同一套参考值，不依赖其他队伍成绩进行归一化。

（二）算例得分与汇总

单种子原始分＝100 ×（40% × WNS 改善比例＋60% × TNS 改善比例）

合法结果的单种子得分为上述原始分与−100 中的较大值，即单种子得分＝max（原始分，−100）。经确认物理运行失败或物理合法性检查不通过的算例与种子组合，单种子得分固定为−100。所有必测组合均纳入计算，每个算例的得分为其三个种子得分的算术平均，最终得分为全部七个算例得分的等权平均。开发阶段的训练与公开验证反馈分别对相应集合中的算例等权平均。

七、功能要求

Harness 应能从统一入口启动，生成可应用的算法修改，并按评估接口返回的信息继续运行。候选算法确定后，应能在不人工修改输入或结果的条件下完成算法执行。

最终候选允许包含切分平衡等物理合法性检查不通过或物理运行失败的算例与种子组合，失败结果参考前述的罚分机制。评估记录保留各组合的合法性状态、失败原因及日志。代码修改范围、函数接口及评估文件完整性等提交要求仍须强制满足，违反这些要求的提交不会以固定罚分替代合规检查。演化执行过程中无效或被拒绝的候选程序也会计入评估次数。

八、开发环境

官方评测平台基于 Open3DBench 和 OpenROAD 开发。组委会提供包含 Python、PyTorch、DREAMPlace、OpenROAD、三维芯片设计流程初始代码及调用示例的统一开发容器。正式评测在统一计算资源配置下执行。

参赛者可在同一设备及相同评估配置下比较候选相对于本机基线的改善。正式成绩仍由统一评测平台使用官方参考值计算。运行方法及环境要求见发布包中的 README。

九、成绩评价

（一）初赛成绩

初赛成绩＝七个算例得分的等权平均（包括三个训练算例、两个公开验证算例和两个隐藏算例）

本赛题仅设初赛，成绩依据全部算例的时序改善情况来计算。不设答辩评分，技术报告用于核实方法与结果，不另行计分。

（二）复核与名次确定

组委会统一复跑参赛队提交的最终 harness，每次复跑均从官方初始算法开始。算法成绩依据正式运行中生成的算法来评测。参赛队提交的算法代码、结果与日志用于复现、提前核验和初步成绩筛选。针对模型 API 调用的随机性，组委会将采用多次复跑取最优结果的策略。

正式成绩经统一评测与复核后公布。同分时，优先比较 TNS 的改善比例。名次仍无法区分时，由组委会复核裁定。

十、解题思路

参赛者可先分析初始算法的关键路径、跨层连接和面积分配，再围绕可解释的假设组织算法演化。例如，将时序关键性引入连接代价，但二维参考中的关键连接未必仍是三维设计中的瓶颈，应结合最终物理反馈判断改动是否有效。

十一、赛题约束条件

（一）评估隔离与代码边界

算法改动与官方评估逻辑相互隔离。候选可以改变允许的设计决策，不得修改评估算法、指标提取与计分程序、参考数据、预算记录，以及违例定义和结果有效性判定。输入网表、工艺条件、时钟约束等输入数据及约束不得篡改。

（二）外层评估循环与预算

组委会在参赛队提交的 harness 之外，再设置一层官方评估循环。外层循环负责启动和监控参赛 harness，通过统一模型调用代理记录 API 用量与人民币费用，通过评估接口核验候选算法并登记评估次数。参赛 harness 负责搜索决策，不能修改、绕过或重置外层计量。

每队每条基线的单次正式运行分别设置最多 10 次候选评估提交和 15 元人民币的模型 API 费用上限，两条基线的额度独立计算。最终提交仍选择其中一条基线。费用以外层 harness 的统一计量记录为准。评估次数耗尽后不再发起新的模型调用或候选评估。模型费用耗尽后停止新的模型调用，已生成的候选仍可在剩余评估额度内提交。

一次候选评估包含三个训练算例与三个种子，共 9 组物理流程运行。因修改范围、函数接口或其他合法性问题被拒绝的提交，以及训练或物理运行失败的候选，均计入评估次数。

正式运行中的所有大模型调用均须经过官方代理，使用指定的 DeepSeek-V4.1-Flash 版本，不得更换或混用其他基模。

（三）数据与过程要求

不得通过算例名称、隐藏输入身份或参考答案硬编码结果，不得在正式运行中人工干预算法生成与选择。公开论文和开源代码可作为方法参考，使用位置、版本及许可证须在材料中说明。

（四）离线专家知识

本赛题旨在考察智能体在统一资源预算下，自主分析问题、修改代码并依据评估反馈持续优化的能力。为减少离线试验积累对成绩可比性的影响，同时保留少量通用领域经验作为优化起点，现对离线专家知识的注入作如下规定。

1. 统一在初始提示词中声明。参赛队伍提供的离线专家知识，必须在初始提示词中集中写明，并清楚标识该部分内容。

2. 严格限制篇幅。总篇幅不超过 200 个计数单位。每个汉字计 1 个单位，每个英文单词计 2 个单位，连续数字计 1 个单位。标点、空格和换行不计入。

3. 仅允许纯文本通用经验。内容必须以自然语言表述，不得包含代码、伪代码或数学公式，不得包含针对具体算例、种子的离线调参结果或预先生成的候选方案。不得通过其他提示词、附加文件、代码注释或其他载体，额外注入或变相隐藏离线

专家知识。智能体在正式运行中根据赛题材料、代码和评估反馈自主形成的经验，不属于上述离线注入内容。

4. 评测时开展合规核验。组委会将对高分队伍的初始提示词、智能体实现、相关文件及运行记录进行检查。经核验发现违反上述约束，包括超出篇幅或通过其他方式绕行注入专家知识的，将认定为不合法提交并取消成绩。

十二、参考文献

[1] He, T. A time scaling theory for multi-layer electronic systems. Science China Information Sciences, 69: 183401, 2026. DOI: 10.1007/s11432-026-5028-5.

[2] He, T. Huawei’s τ Chip Was Supposed to Melt? arXiv:2609.04287, 2026.

[3] Shi, Y., Gao, C., et al. Open3DBench: Open-Source Benchmark for 3D-IC Backend Implementation and PPA Evaluation. arXiv:2503.12946v2, 2026.

[4] Lu, Y.-C., et al. TP-GNN: A Graph Neural Network Framework for Tier Partitioning in Monolithic 3D ICs. Design Automation Conference, 2020.

[5] Lin, Y., et al. DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement. Design Automation Conference, 2019.

[6] OpenROAD Project. OpenROAD Documentation.

[7] Novikov, A., et al. AlphaEvolve: A Coding Agent for Scientific and Algorithmic Discovery. arXiv:2506.13131, 2025.

[8] Zhang, J., et al. Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents. arXiv:2505.22954, 2025.

十三、提交要求

（一）需提交的技术材料

每队选择一个开发基线，提交最终 harness、生成的算法代码及对应切分和布局结果，相关作品材料整理至同一文件夹，材料统一打包为 ZIP（.zip）压缩文件，命名为“参赛编号_赛题 2.zip”，上传至**“作品材料文件夹百度网盘链接”**入口，请参赛选手认真检查链接的有效性，分享链接可设置为永久有效，确保评委能够顺利访问材料。作品材料检查无误后上传至在线网盘（推荐使用百度网盘，**分享时选择自动填充提取码**）。

1. 说明与清单。README.md 写明所选基线、任务包与模型版本、运行命令和复现步骤；requirements.txt 固定额外的 harness 依赖版本，候选程序的运行环境保持固定。

2. Harness 源码。提交所选基线对应的完整 Agent 开发目录，即 AgenticBench/agent/binary_loss/ 或 AgenticBench/agent/loss_decoder/，包含提示模板、配置和辅助文件。目录中的 agent.py 应实现 Agent.run(context)，返回已完成训练评估的候选 ID。组委会通过外层 harness 加载该入口。

3. 算法代码。提交所选基线的 program.py。概率预测基线包含 compute_binary_partition_loss，表示学习基线包含 compute_tp_gnn_loss 和 decode_tp_gnn_embeddings。可以只修改表示学习基线的其中一个函数，另一个保留初始实现。函数签名、参数顺序和数值接口须与任务包一致。Agent 通过评估接口提交完整的 Python 源码文本，由外层 harness 封装并运行。

4. 切分与布局等物理设计结果。物理设计结果按“子任务/算例/种子”组织，包含切分方案、上下层布局、评测指标和时序报告。合法结果的提交内容应来自通过验证的原生运行结果。失败结果保留评估日志与失败原因，不要求提供未生成的物理结果文件。需要保留物理文件时，应在运行前按 README 启用 --retain-physical-files 选项。

5. 运行日志。logs/ 保存最终 harness 运行时的算法演化记录，包括算法改动的父子关系、代码版本、评估结果和失败原因等信息，供组委会初步核验和筛选。

（二）需提交的文字材料

参赛队需提交技术报告，按附录说明所选基线、harness 设计、最终算法和实验结果。

十四、其他说明

报名、任务发布、提交截止和成绩公示时间以赛事通知为准。奖励、名额、证书及奖金安排以大赛章程和赛题通知为准。

参赛成果须注明论文、开源代码及外部工具的贡献。比赛过程中，参赛队伍可反馈赛题相关的问题，组委会进行答疑与勘误。

十五、联系方式

赛事信息与报名：www.aicomp.cn/notice/notice-3/5484.html。本赛题交流方式与技术支持渠道以正式赛事通知为准。

附录　技术方案大纲

技术报告应围绕所选基线、harness 设计与实验结果展开，可按以下结构组织。

一、算法概述

概述网表切分问题、已有主流方法及所选开发基线，概述 harness 的主要思路和相对初始算法的结果。

二、实现方案

给出 harness 的设计动机、工作流程或伪代码，说明提示词组织、工具调用和反馈分析等算法环节。尝试解释最终损失函数，并分析其性质。采用聚类基线时，说明解码方法及其与学习目标的配合。

三、实验与创新点

列出训练与公开验证算例的完整指标对比，比较初始程序和最终算法的结果，并分析演化效率。可提供关键算法部件的消融实验。

四、问题与思考

分析未改善或运行失败的算例，讨论跨算例适应、搜索稳定性及方法局限。

五、过程进度与团队分工

简述主要版本、迭代过程和预算消耗，列明成员分工、外部指导及工具辅助的作用。

六、复现与参考资源

给出完整复现入口及材料索引，列明论文、代码、软件、模型与数据来源、版本和使用位置。