计 算 机 研 究 与 发 展
Journal of Computer Research and Development

https://crad.ict.ac.cn
63(7): 1643−1669, 2026

本期推荐

人工智能辅助的芯片物理设计方法综述

刘睿之[1,2] 李捷鹏[1,2] 王 睿[1,2] 隋京言[3] 李兴权[4] 解壁伟[1,2,5] 于春功[1,2,6] 卜东波[1,2,6]

[1]（处理器芯片全国重点实验室（中国科学院计算技术研究所） 北京 100190）

[2]（中国科学院大学 北京 100049）

[3]（聊城大学计算机学院 山东聊城 252000）

[4]（东南大学集成电路学院 南京 214026）

[5]（北京开源芯片研究院 北京 100080）

[6]（中原人工智能产业研究院 郑州 450046）

（liuruizhi19s@ict.ac.cn）

AI-Assisted Methods for Chip Physical Design: A Survey

Liu Ruizhi[1,2], Li Jiepeng[1,2], Wang Rui[1,2], Sui Jingyan[3], Li Xingquan[4], Xie Biwei[1,2,5], Yu Chungong[1,2,6], and Bu Dongbo[1,2,6]

[1]（*State Key Lab of Processors* (*Institute of Computing Technology*, *Chinese Academy of Sciences*), *Beijing* 100190）

[2]（*University of Chinese Academy of Sciences*, *Beijing* 100049）

[3]（*School of Computer Science*, *Liaocheng University*, *Liaocheng*, *Shandong* 252000）

[4]（*School of Integrated Circuits*, *Southeast University*, *Nanjing* 214026）

[5]（*Beijing Institute of Open Source Chip*, *Beijing* 100080）

[6]（*Central China Artificial Intelligence Research Institute*, *Zhengzhou* 450046）

Abstract As semiconductor technology scales, chip physical design confronts escalating complexity and multi-objective optimization challenges, with traditional electronic design automation (EDA) tools encountering bottlenecks in computational efficiency and convergence. Artificial intelligence (AI), specifically deep learning (DL) and reinforcement learning (RL), offers a transformative paradigm for data-driven optimization in physical design. This survey systematically reviews the application of AI methods in the four core stages of chip physical design: floorplanning, placement, clock tree synthesis, and routing. First, from the perspective of optimization objectives, representative AI methods and their technical contributions in each stage are summarized. Second, a critical analysis of the gap between academic datasets and industrial scenarios is conducted, revealing key data bottlenecks. Third, the integration approaches between AI methods and traditional EDA flows across different stages are examined. Then, the evolutionary trends of AI methods in each stage are systematically reviewed, highlighting the paradigm shift from isolated, geometry-driven optimization toward cross-stage collaboration and AI-native chip physical design. Although AI methods have significantly improved design efficiency and quality, their widespread adoption is still limited by data scarcity, insufficient model interpretability, and challenges in integration with existing toolchains. Finally, we discuss the limitations of existing AI-driven approaches in physical chip design and outline promising research directions, including open-source AI-EDA infrastructure, large circuit foundation models, and AI-native EDA.

Key words chip physical design; artificial intelligence (AI); deep learning; reinforcement learning; electronic

收稿日期：2025-12-31；**修回日期**：2026-06-04

基金项目：国家重点研发计划（2020YFA0907000）；国家自然科学基金项目（32270657, 32271297, 82130055, 62072435）

This work was supported by the National Key Research and Development Program of China (2020YFA0907000) and the National Natural Science Foundation of China (32270657, 32271297, 82130055, 62072435).

通信作者：卜东波（dbu@ict.ac.cn）

design automation (EDA)

摘 要 随着半导体技术的发展,芯片物理设计面临日益增长的复杂度与多目标优化挑战,传统电子设计自动化(electronic design automation, EDA)工具在效率与收敛性上面临瓶颈。人工智能(artificial intelligence, AI)技术,特别是深度学习与强化学习,为数据驱动的设计优化提供了新范式。系统综述了AI方法在芯片物理设计的4个核心阶段:版图规划、布局、时钟树综合与布线中的应用进展。首先,从优化目标角度,梳理了各阶段代表性AI方法及其技术贡献;其次,批判性分析了学术数据与工业级场景间的差异,揭示了关键数据瓶颈;再次,评述了不同阶段AI方法与传统EDA流程的集成方式;而后,系统总结了各阶段AI方法的演进趋势,指出从单点优化、几何驱动向跨阶段协同、AI原生设计的范式转变;尽管AI方法已显著提升设计效率与质量,其广泛应用仍受限于数据稀缺、模型可解释性不足以及与现有工具链的集成挑战,因此在最后,分析现有芯片物理设计AI方法的局限性,并展望开源AI-EDA基础设施、电路大模型、AI原生EDA等未来研究方向。

关键词 芯片物理设计;人工智能;深度学习;强化学习;电子设计自动化

中图法分类号 TP183

DOI: 10.7544/issn1000-1239.202660001 **CSTR:** 32373.14.issn1000-1239.202660001

随着半导体技术的迅猛发展,超大规模集成电路(very large scale integration, VLSI)的规模和复杂性呈指数级增长。芯片制程节点已从传统的微米级演进至当前的纳米级(如7 nm、5 nm乃至3 nm及以下),这不仅带来了更高的晶体管密度和性能提升,也显著加剧了电子设计自动化(electronic design automation, EDA)工具在芯片设计流程中的挑战[1]。在物理设计阶段,工程师面临着海量数据处理、优化算法的计算复杂性以及芯片功耗、性能和面积(power, performance, area, PPA)的多目标权衡问题[2-3]。传统EDA工具依赖于启发式算法和规则驱动方法,在处理现代芯片的异构架构、多核并行和先进工艺节点时,往往陷入效率瓶颈或局部最优陷阱[1,4]。例如,在版图规划和布线阶段,信号完整性问题和拥塞预测精度不足已成为制约设计收敛的关键因素,导致迭代次数增加和设计周期延长[2]。

在这一背景下,人工智能(artificial intelligence, AI)开始成为推动EDA演进的重要力量。深度学习、强化学习和生成式模型已不再只是辅助分析工具,而是逐步参与设计优化的关键决策过程,能够借助数据驱动机制自动发现模式,预测优化路径并加速求解[4]。在芯片物理设计中,AI的应用已广泛渗透到后端流程的多个环节,例如使用神经网络预测布局拥塞[5],强化学习优化宏单元放置[3],以及利用学习方法生成或优化布线拓扑[6-8]。这些方法不仅提升了设计效率,还在部分场景下展现出优于传统工具的潜力,推动了芯片设计范式从规则驱动向数据驱动转变。然而,尽管AI在EDA中的潜力巨大,其实际部署仍面临数据稀缺、模型泛化性和鲁棒性以及与传统工具集成的挑战[1,4,9-10],这也激发了学术界和工业界的持续探索。

本综述聚焦于AI驱动的芯片后端物理设计阶段,具体限定在版图规划(floorplanning)、布局(placement)、时钟树综合(clock tree synthesis, CTS)和布线(routing)四个核心子阶段,并将其作为4个独立章节加以讨论。主要因为该阶段是芯片设计流程中计算最密集、优化最敏感的环节之一,直接影响最终芯片的PPA指标和制造良率[11-12],相比之下,前端设计(如RTL综合)更多涉及逻辑抽象,与物理实现的关联相对间接[13]。受篇幅所限,本综述并未将时序优化(timing optimization)作为单独的章节探讨,但考虑到其贯穿整个物理设计流程[14],我们在各节中均纳入了与时序驱动相关的研究。另一方面,近年来针对芯片前端设计与逻辑综合阶段的AI综述已较为丰富[15-16],而面向物理设计的综述要么发布时间较早[2,17-18],要么仅聚焦于特定设计阶段(如布线)[19-22],尚不足以覆盖近两三年的研究进展及其方法学变化,这也是本综述选题的主要原因。

与已有综述相比,本文的增量贡献主要体现在4个方面。1)本文并非仅按设计阶段或模型类型罗列文献,而是从优化目标与应用场景出发,对各阶段AI方法进行统一分类,以凸显问题定义与算法设计之间的对应关系。2)本文系统梳理了各阶段文献所依赖的数据基准,并重点比较学术公开数据与真实

工业场景在设计规模、物理约束、签核闭环和开源可得性上的差异，从数据层面分析模型泛化受限的根源。3)本文专门讨论了 AI 算法与传统 EDA 工具的结合方式，区分点工具替代、嵌入式指导、闭环联合优化等不同集成模式，以回应学术界和工业界共同关心的落地问题。4)本文在文献梳理基础上总结了芯片物理设计 AI 方法从单点优化、几何驱动向跨阶段协同、语义驱动和 AI 原生设计演进的整体趋势。基于上述归纳框架，本文试图为研究者和从业者提供一个兼顾方法、数据与系统集成视角的综述框架。帮助研究者和从业者更好地了解 AI 驱动芯片物理设计的发展趋势。

1 相关概念

本节首先简要介绍芯片后端物理设计的整体流程，并重点说明版图规划、布局、时钟树综合和布线 4 个核心阶段的问题定义。随后，仅围绕后文分析直接相关的 AI 基础展开概述，包括常用表示学习模型以及监督学习、强化学习和自监督学习等主要范式，为后续讨论 AI 在芯片物理设计中的应用提供必要背景。

1.1 芯片后端物理设计流程

芯片设计流程通常被划分为前端设计(frontend design)与后端设计(backend design)。前端设计主要关注电路的逻辑功能实现；而后端设计，即物理设计(physical design)，则致力于将逻辑网表(netlist)转换为可供制造的几何版图(layout)[12]。物理设计是决定芯片最终 PPA 的关键环节[14]。本文所探讨的物理设计流程主要包含 4 个核心阶段：版图规划、布局、时钟树综合(CTS)以及布线。这 4 个阶段按顺序执行，各阶段的优化结果紧密耦合，共同决定了最终设计的质量[11]。典型芯片设计流程和物理设计流程如图 1 所示。

1.1.1 版图规划

版图规划是物理设计的起始阶段，也是对芯片 PPA 影响深远的阶段之一[14]。在这一阶段，设计者需要确定芯片的整体尺寸与形状，规划输入输出(I/O)单元的位置，并确定宏模块(macro，如存储器、IP 核等)在芯片核心区域内的摆放位置与方向[14,23]。

给定一个模块集 $M=\{m_1,m_2,\cdots,m_n\}$，其中每个模块 m_i 具有固定的面积 A_i 以及允许的纵横比范围 $[r_{\min},r_{\max}]$。版图规划的目标是确定每个模块 m_i 的左下角坐标 (x_i,y_i) 及其形状，以最小化加权代价函数。通常，目标函数(*Cost*)由芯片总面积(*Area*)和模块间的总互连线长(*Wirelength*)组成：

$$Cost=\alpha\cdot Area+\beta\cdot Wirelength,$$

其中，α 和 β 为权重系数。该优化问题需满足模块间的不重叠约束。

此外，版图规划还需考虑电源网格规划(power planning)及 I/O 布局等约束。版图规划的结果直接决定了后续标准单元的可放置区域以及全局布线拥塞度。

1.1.2 布局

布局阶段的主要任务是将网表中的数百万个标准单元(standard cells)放置到版图规划预留的行(row)区域内。布局质量直接影响芯片的时序收敛性和可布线性[17,24]。

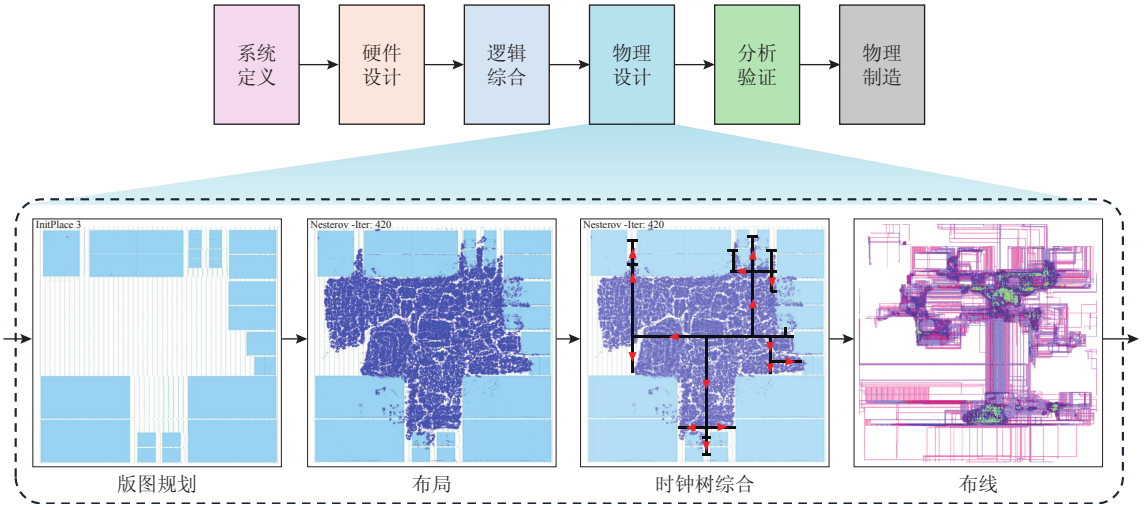


Fig. 1 Typical flow of chip design and the physical design process

图 1 典型芯片设计流程及物理设计流程

给定标准单元集合$C=\{c_1,c_2,\cdots,c_k\}$和网表连接关系$N=\{n_1,n_2,\cdots,n_m\}$, 布局问题的目标是确定每个单元c_i的坐标(x_i,y_i), 在满足密度约束的同时最小化总线长, 以避免局部过度拥塞。业界通常使用半周长线长(half-perimeter wire length, *HPWL*)作为线长的近似估算指标。对于一个连接多个引脚的线网$n\in N$, 其*HPWL*定义为包含该网所有引脚的最小边界框(bounding box)的半周长:

$$HPWL_n=\left(\max_{i\in n}x_i-\min_{i\in n}x_i\right)+\left(\max_{i\in n}y_i-\min_{i\in n}y_i\right),$$

总的基础优化目标为最小化所有线网的*HPWL*之和:

$$\min\sum_{n\in N}HPWL_n。$$

同时需满足密度约束, 即在任意给定的区域(bin)b内, 单元的总面积不得超过该区域的容量$capacity_b$:

$$\sum_{c_i\in b}Area(c_i)\leqslant capacity_b。$$

布局通常分为全局布局(global placement)和详细布局(detailed placement)两个子阶段。传统全局布局通过松弛约束求解大致位置, 而详细布局则将单元对齐到合法的格点上并消除重叠。

1.1.3 时钟树综合

时钟树综合的目标是构建一个将时钟源(source)信号传递到芯片中所有时钟汇点(sink)的互连网络[14]。

给定时钟源s_0和时钟汇点集合$S=\{s_1,s_2,\cdots,s_p\}$, 时钟树综合旨在构建一个缓冲器(buffer)树或网格结构。其核心优化指标包括时钟偏斜(*Skew*)和时钟延迟(*Latency*)。对于任意2个汇点s_i和s_j, 时钟偏斜定义为时钟信号到达时间的差值:

$$Skew=\max_{i,j\in S}|t_{\mathrm{arr}}(i)-t_{\mathrm{arr}}(j)|,$$

其中$t_{\mathrm{arr}}(i)$表示时钟信号从源传导到汇点s_i的延迟。时钟延迟定义为时钟信号从源到达任意汇点的最大延迟:

$$Latency=\max_{i\in S}t_{\mathrm{arr}}(i)。$$

时钟树综合的目标通常是在满足给定的最大偏斜约束$Skew_{\mathrm{target}}$的前提下, 最小化功耗(*Power*)(由插入的缓冲器数量和总线长决定)和延迟:

$$\min(Power+\gamma\cdot Latency),$$

$$\text{s.t. } Skew\leqslant Skew_{\mathrm{target}}。$$

时钟树的质量对芯片的主频和功耗有决定性影响。理想的时钟树应具有零偏斜(zero skew)或特定的有用偏斜以优化时序。

1.1.4 布线

布线是物理设计的最后一步, 旨在利用金属层将网表中的所有信号连接起来, 同时满足设计规则检查(design rule check, DRC)[14,18]。

布线问题通常被建模为在多层网格图$G(V,E)$上的路径查找问题。它分为全局布线(global routing)和详细布线(detailed routing)。全局布线将芯片划分为一系列矩形区域(G-Cells), 并为每个网n_j寻找通过这些区域的路径, 通常通过构建Steiner树(Steiner tree)来实现。其目标是最小化总拥塞(congestion)和线长。拥塞定义为经过某条边界的连线数量超过了该边界可容纳的布线容量(capacity)。对于每条边$e\in E$, 设$demand(e)$为经过该边的连线数, $capacity(e)$为该边的容量, 则全局布线试图最小化:

$$Cost=Total\ wirelength+\sum_{e\in E}\max(0,demand(e)-capacity(e))。$$

详细布线则在全局布线确定的路径指导下, 具体分配金属线和通孔(via), 确保满足最小线宽、最小间距等DRC规则。

1.1.5 简化问题模型与真实工业约束的差异

1.1.1~1.1.4节所述的版图规划、布局、时钟树综合和布线问题的数学表述, 主要用于给出各阶段最基本的优化对象与目标函数, 便于后文统一讨论。然而, 在真实工业流程中, 物理设计并不是若干彼此独立的单目标优化问题, 而是一个跨阶段、强反馈、多约束耦合的系统工程。以布局和布线为例, 实际优化不仅要考虑线长、密度或拥塞, 还往往同时受到时序、宏模块阻塞(macro blockage)、引脚可达性(pin accessibility)、电源配送网络(power delivery network, PDN)及拥塞与设计规则违例之间的耦合关系等因素制约。类似地, 时钟树综合阶段的偏斜与延迟优化也会受到后续布线寄生、缓冲器合法化和功耗预算的反向影响。

此外, 工业级物理设计流程通常并非一次性完成, 而是伴随工程变更指令(engineering change order, ECO)迭代、签核驱动修正以及前后阶段之间的多轮反馈。例如, 早期布局阶段最小化的半周长线长并不必然对应最终详细布线后的最优时序或最少DRC; 全局布线中的低拥塞结果也不一定能够直接转化为详细布线阶段的高可布线性。换言之, 许多学术研究中常用的目标函数更多是代理目标(proxy objectives), 而非完整工业签核流程中的最终目标。正因如此, 后文对AI方法的讨论将不仅关注其在某一阶段代

理指标上的提升，还将特别考察这些方法如何处理真实约束，如何与 EDA 流程形成反馈闭环，以及它们与最终 QoR 之间的一致性程度。

1.2 AI 技术基础

随着半导体工艺的演进，芯片物理设计问题的搜索空间迅速膨胀，往往呈现出高维、离散、多约束耦合的特征，使传统的启发式算法面临巨大挑战。近年来，AI 被广泛引入以处理这类复杂优化问题，而机器学习（machine learning）致力于通过从数据中解析模式来构建预测模型或决策策略。在深度学习流行之前，一系列经典的机器学习算法，如逻辑回归（logistic regression）[25]、支持向量机（support vector machine, SVM）[26]、聚类（clustering）[27]、决策树与随机森林（decision tree & random forest）[28]、朴素贝叶斯（naive Bayes）[29]以及梯度增强算法（gradient boosting algorithms）[30-31]等算法，已被广泛研究并应用于各类物理设计流程中的数据分析任务。考虑到本文的重点在于 AI 方法如何服务于芯片物理设计，本节不再对通用机器学习算法作展开介绍，而仅概述与后续章节联系最紧密的几类模型与学习范式。

1.2.1 与芯片物理设计相关的深度学习代表性模型

深度学习利用多层神经网络（neural networks）[32]，并使用反向传播机制[33]来学习数据的高阶特征表示。以下介绍 4 种在物理设计中具有潜在应用价值的代表性网络架构。

1）全连接网络（fully connected network, FCN）。全连接网络，也称为多层感知机（multilayer perceptron, MLP），是最基础的神经网络形式[32]。网络由输入层、若干隐藏层和输出层组成，层与层之间的神经元完全连接。对于第 l 层，其输出 $\boldsymbol{h}^{(l)}$ 可表示为：

$$\boldsymbol{h}^{(l)}=\sigma(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)}+\boldsymbol{b}^{(l)}),$$

其中 $\boldsymbol{W}^{(l)}$ 和 $\boldsymbol{b}^{(l)}$ 分别为权重矩阵和偏置向量，$\sigma(\cdot)$ 为非线性激活函数（如 ReLU, Sigmoid）。

2）卷积神经网络（convolutional neural network, CNN）。在芯片物理设计中，布局图、拥塞图、功耗图等具有规则网格结构的输入，CNN 及其变体常被用于提取局部几何特征，并广泛应用于拥塞预测、可布线性分析与热图回归等任务[5,34]。CNN 的一般结构如图 2 所示。其核心是卷积操作，通过卷积核（kernel）提取局部特征。二维卷积操作可定义为[35]：

$$(\boldsymbol{I}*\boldsymbol{K})(i,j)=\sum_m\sum_n I(i+m,j+n)K(m,n),$$

其中 $\boldsymbol{I}$ 为输入图像或特征图，$\boldsymbol{K}$ 为卷积核。CNN 具有平移不变性。使用 CNN 进行可布线性预测的网络结构如图 3 所示。

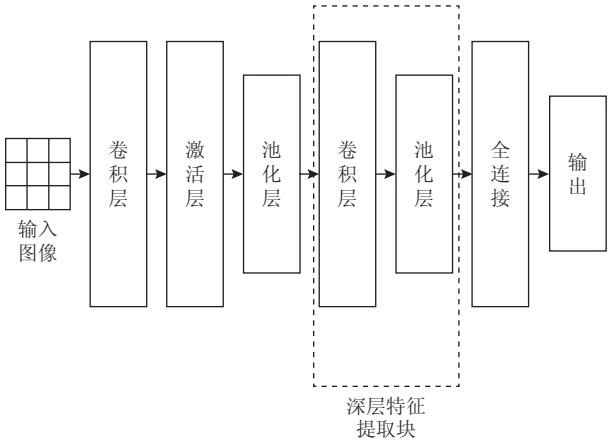


Fig. 2 Typical structure of CNN

图 2 典型卷积神经网络结构

3）图神经网络（graph neural network, GNN）。对于电路网表、时序路径和超图连接关系等非欧几里得结构，GNN 如图 4 所示，能够通过消息传递机制聚合节点与边的邻域信息，因此成为建模电路拓扑、关键路径依赖和跨模块交互的核心工具[3,22,36]。GNN 通过聚合邻居节点的信息来更新当前节点的特征表示。通用的消息传递（message passing）公式为：

$$\boldsymbol{h}_v^{(l+1)}=U^{(l)}\left(\boldsymbol{h}_v^{(l)},AGG^{(l)}(\{\boldsymbol{h}_u^{(l)},\forall u\in\mathcal{N}(v)\})\right),$$

其中 $\boldsymbol{h}_v^{(l)}$ 表示节点 v 在第 l 层的特征，$\mathcal{N}(v)$ 为邻居节点集合，$AGG^{(l)}$ 代表消息聚合操作，$U^{(l)}$ 代表消息更新函

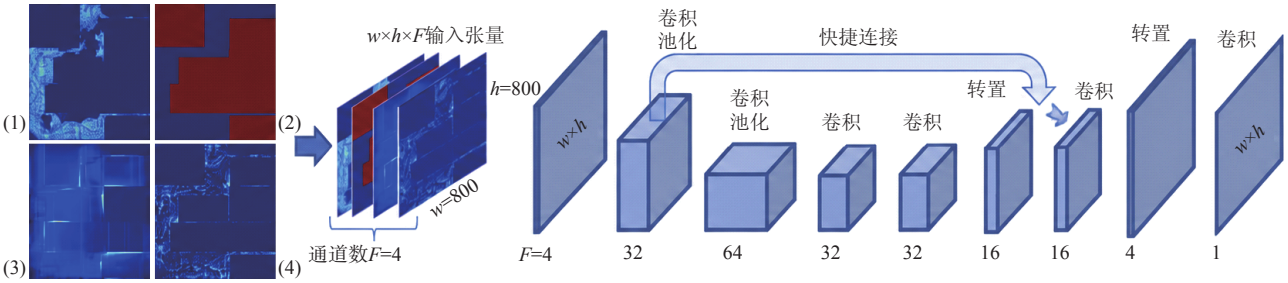


Fig. 3 Prediction of routability using CNN[5]

图 3 使用卷积神经网络进行可布线性预测[5]

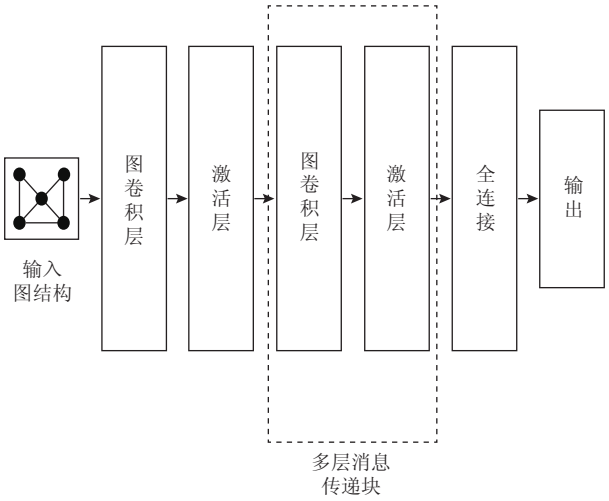


Fig. 4 Typical structure of GNN

图 4 典型图神经网络结构

数。每一轮消息聚合和更新都会使得图中每一个节点获得其邻居节点的信息。GNN 用于版图尺寸预测如图 5 所示。

4)注意力机制与 Transformer 网络。Transformer 架构基于注意力(attention)机制,注意力机制最初用于提升序列到序列模型的性能,允许模型在生成输出时动态关注输入的不同部分[38-39]。Transformer 摒弃了循环和卷积[40],加入位置编码和多头注意力机制以增强对序列结构的建模能力和并行计算能力,结构如图 6 所示。其核心注意力计算为:

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V},$$

其中 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ 分别为查询(query)、键(key)和值(value)矩阵,d_k 为键的维度。Transformer 及其注意力机制擅长建模长程依赖关系,适合处理拓扑生成、序列决策和跨阶段上下文编码等问题,近年来在宏单元布局、Steiner 树生成及电路表征学习中逐步受到重视,将其应用于混合尺寸布局的网络结构如图 7 所示。

1.2.2 常见学习范式及其在物理设计中的角色

根据数据标签的可获得性和反馈机制,AI 方法的应用模式主要分为 3 类:

1)监督学习(supervised learning)。监督学习利用

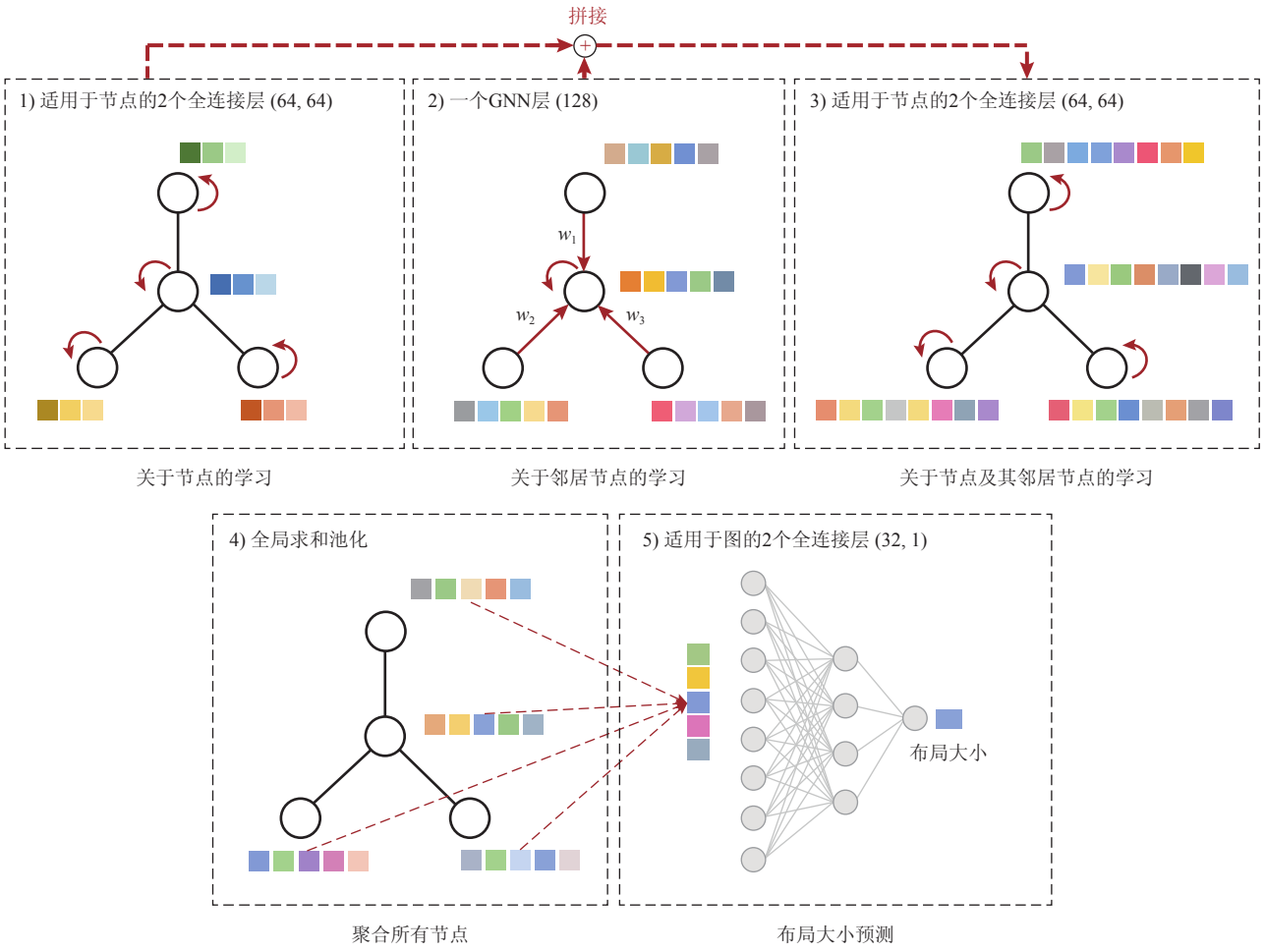


Fig. 5 Layout size is predicted by GNN[37]

图 5 图神经网络用于版图尺寸预测[37]

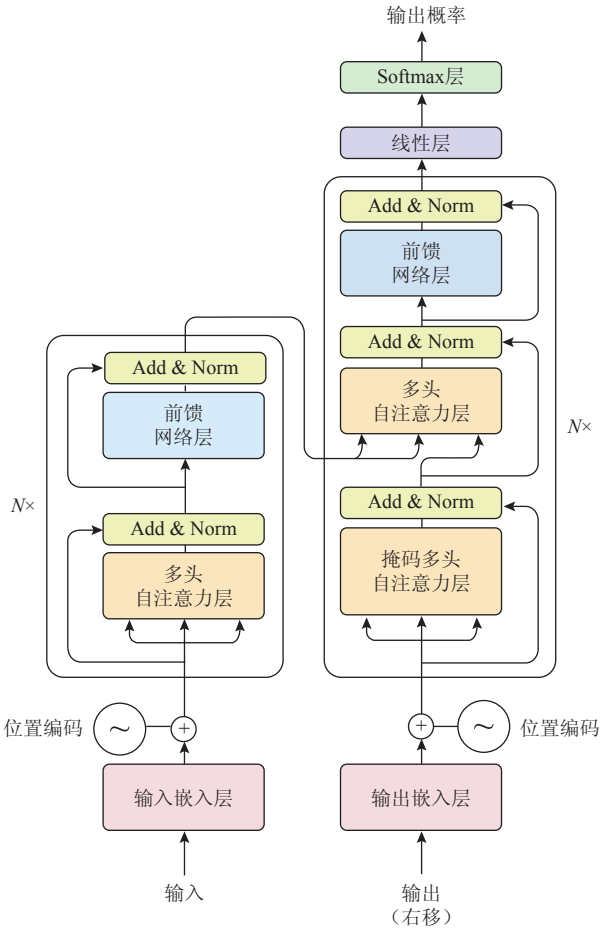


Fig. 6 Typical network structure of Transformer[40]

图 6 典型 Transformer 网络结构[40]

包含输入-输出对的标注数据集进行训练，旨在学习一个从输入空间$\mathcal{X}$到输出空间$\mathcal{Y}$的映射函数f_θ[32]。给定训练数据集$\mathcal{D}=\{(x_1,y_1),\cdots,(x_N,y_N)\}$，目标是最小化损失函数$\mathcal{L}$：

$$\min_\theta \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(f_\theta(x_i),y_i)\ 。$$

在物理设计中，监督学习这类方法常用于预测拥塞、时序、功耗、设计规则违例或可布线性等中间指标，从而替代部分昂贵的分析流程。

2）强化学习。强化学习关注智能体（agent）如何在环境（environment）中采取行动以最大化累积奖励[3]。它非常适合解决物理设计中的序列决策问题。问题通常建模为马尔可夫决策过程（MDP），目标是学习策略$\pi(a|s)$以最大化期望累积回报G_t：

$$G_t=\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\ 。$$

通过与 EDA 工具交互，强化学习智能体可以探索出超越人类经验的策略。

3）自监督学习（self-supervised learning）。自监督学习与预训练范式，不依赖人工标注，而通过电路数据自身的结构构造训练目标，学习可迁移的表示。这一类方法近年来在跨设计泛化、离线策略学习以及大模型构建中显示出越来越强的潜力。大语言模型（LLMs）是自监督学习的典型代表，通常通过“下一个词预测”任务进行预训练[42]。给定序列$W=\{w_1,\cdots,w_t\}$，目标最大化条件概率：

$$\max_\theta \sum_t \log P(w_{t+1}|w_1,\cdots,w_t;\theta)\ 。$$

需要强调的是，上述模型与范式只是芯片物理设计 AI 研究的方法基础。后续章节涉及的大多数

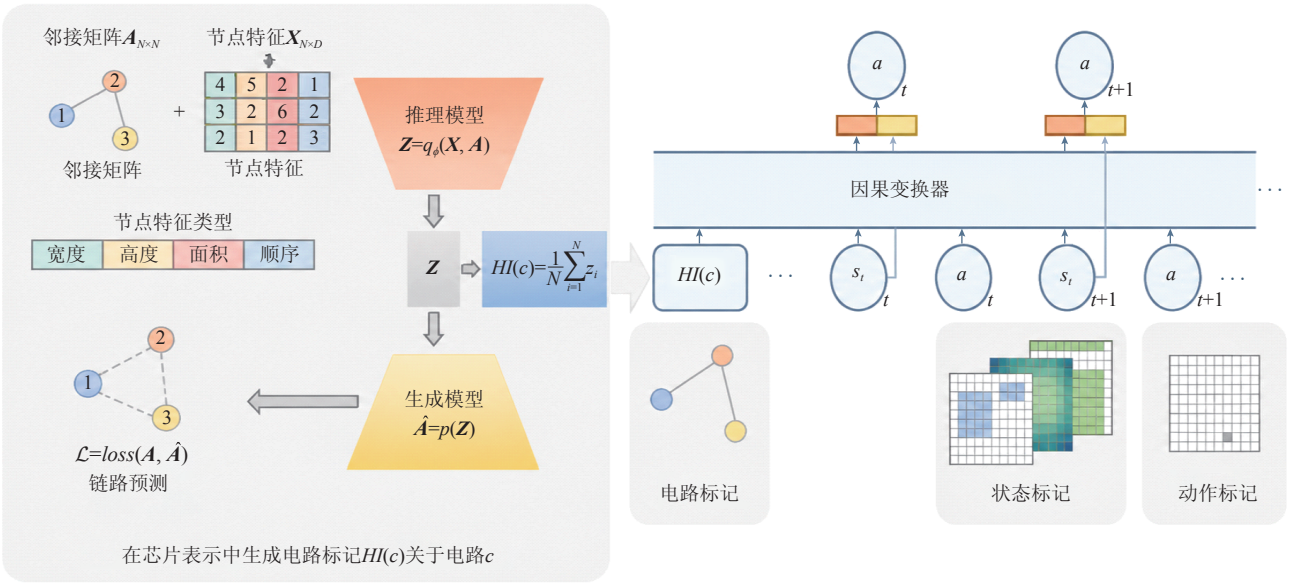


Fig. 7 Mixed-size placement using Transformer[41]

图 7 Transformer 网络用于混合尺寸布局[41]

工作并非简单套用通用 AI 模型，而是围绕电路拓扑、物理约束、EDA 流程反馈和签核目标，对模型结构、损失函数与训练流程进行了有针对性的领域化改造。因此，与其泛泛掌握模型定义，更重要的是理解它们在芯片场景中的任务化与工程化方式。

如图 8 所示，该图直观地展示了不同类型的 AI 方法在芯片物理设计的不同阶段的结合方式和演进路径。随着 AI 方法的不断发展以及先进工艺节点对效率和精度的要求的不断提升，物理设计中的 AI 方法经历了从早期效果预测器和评估器到如今闭环优化与 AI 原生物理设计算法的演化路径，其中的代表性工作以及方法的详细介绍将在后文展开。

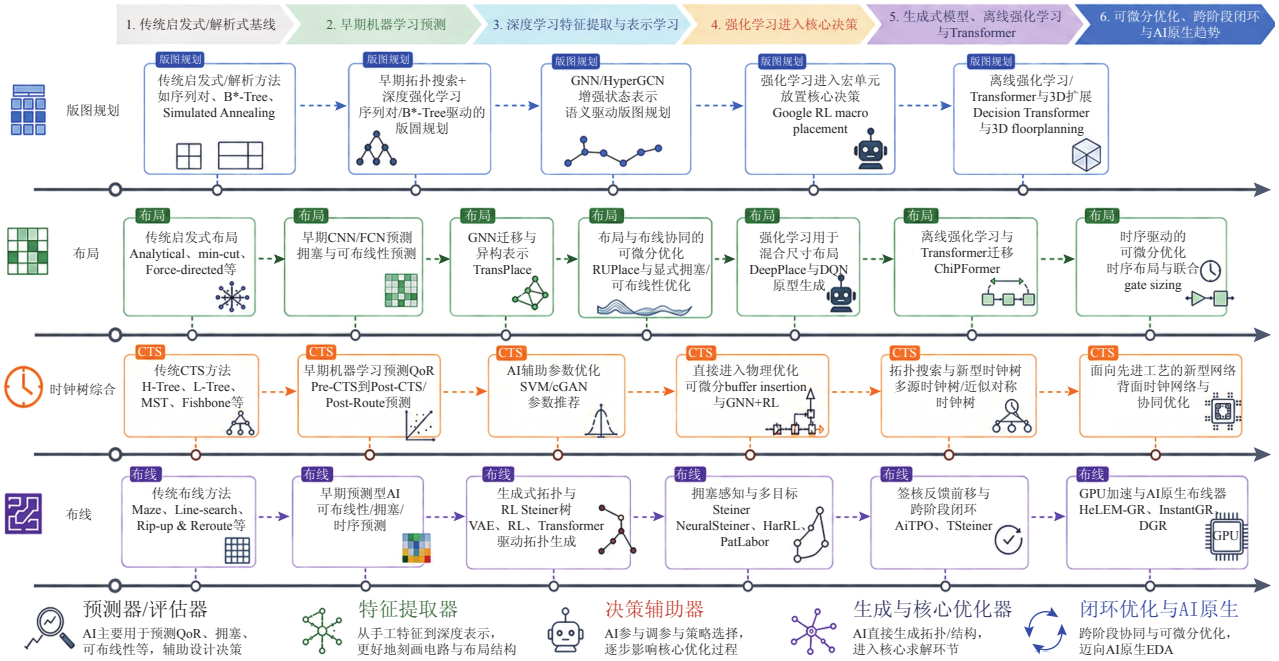


Fig. 8 Evolution paths of AI methods in chip physical design

图 8 芯片物理设计中的 AI 方法演进路径

2 版图规划问题的 AI 求解算法

版图规划作为物理设计的首要环节，其决策质量深刻影响了芯片的 PPA 乃至后续布局和布线阶段的可行性[14]。由于版图规划本质上是一个具有巨大解空间的 NP-hard 组合优化问题，近年来，随着 AI 技术的飞速发展，越来越多的研究开始探索利用 AI 尤其是强化学习和 GNN 来突破传统方法的局限。本节将从应用场景分类、数据挑战、工业结合度及 AI 方法演进等维度，对版图规划领域的 AI 方法进行全面综述。

2.1 基于优化目标与应用场景分类

针对版图规划中不同的设计目标与约束条件，现有的 AI 研究主要可划分为以下 3 个方向。

2.1.1 二维版图规划与混合尺寸规划

二维芯片的版图规划研究是该领域最为基础且广泛研究的方向，AI 方法的引入旨在高效求解模块的非重叠排列以最小化面积和线长[21]。该类研究的核心在于将连续几何优化问题转化为离散组合优化问题。Yu 等人[43] 采用了经典的“序列对”(sequence pair)表示，结合深度强化学习解决模块布局问题。Li 等人[44] 和 Li 等人[45] 则分别利用 B*-Tree 及其变体 SCB-Tree，通过进化算法与强化学习的协同或端到端的深度学习框架，克服了传统模拟退火算法收敛速度慢、易陷入局部最优的缺陷。此外，Amini 等人[46] 引入了“角块列表”(corner block list)表示，进一步丰富了拓扑优化的手段。Mirhoseini 等人[3] 利用基于强化学习的智能体直接放置宏单元进行版图规划，再结合力导向算法处理标准单元，实现了宏单元和标准单元混合尺寸布局的自动化。Shi 等人[47] 针对混合尺寸布局中的时序优化难题，提出了递归原型构建策略，有效缓解了宏单元与标准单元在解析法优化中的相互干扰。Liu 等人[48] 则侧重于通过学习电路连通性与物理位置的映射关系，为下游布局器提供高质量的初始解，而非替代整个版图规划流程。

2.1.2 三维版图规划与特定物理约束的规划

摩尔定律的放缓推动了芯片设计向三维堆叠和

更复杂的物理约束演进。Zhong 等人[49]和 Amin 等人[50]针对 3D IC 布局规划提出了创新方案，重点解决了层分配（layer assignment）与跨层对齐（alignment）这一核心难题。Zhong 等人[49]通过多模态表示和异步层决策机制，实现了对齐约束的有效处理。在模拟电路领域，Basso 等人[20]开发了布线感知的自动布局引擎，专门优化了布局对布线寄生参数和对称性的敏感性。

2.1.3 辅助预测与数据集构建

部分研究致力于解决布局规划中的预测与数据短缺问题。Ji 等人[37]利用 GNN 实现了高精度的模块版图尺寸预测，为早期设计规划提供了重要参考。针对高质量工业级数据匮乏的痛点，Mallappa 等人[23]构建了包含百万级模块且带有真实工业约束的大规模合成数据集 FloorSet，为 AI 算法的评估提供了标准化的基准。

2.2 数据基准与实际工业级应用场景差异

数据是验证算法有效性的核心。当前版图规划的 AI 算法相关的文献在数据来源和规模上呈现出显著的学术界与工业界分化。

2.2.1 版图规划学术数据基准

大部分学术论文[43-46,49-53]仍主要依赖 MCNC 和 GSRC 等发布较早的基准数据集，如表 1 所示。这些数据集模块数量少（几十至几百个），缺乏现代工艺约束（如多电压域、复杂 DRC），导致基于此得出的结论在面临现代大规模 SoC（数千宏单元、亿级标准单元）时往往难以复现，相关文献使用的开源数据集见表 1。

Table 1 Open-source Benchmarks for AI-driven Floorplanning Algorithm

表 1 版图规划 AI 算法相关开源数据集

数据集	发布时间	数据集简介	使用该数据集的相关文献
MCNC	1996	早期开源的版图规划数据集，主要包含中小型电路，模块个数 10~50	[43−46, 49−50, 52−53]
GSRC	2000	针对大规模版图规划问题的合成数据，模块个数 10~300	[43−46, 49−50, 52−53]
ISPD 2005	2005	来自 ISPD2005 布局比赛的公开数据，包含从 20 万到超过 200 万可移动模块的不同规模电路	[48, 54]
FloorSet	2025	大规模合成数据集，基于真实商业 SoC 的统计分布生成，包含复杂的硬性设计约束	[48]
iDATA	2025	基于 AiEDA 库生成的大规模结构化数据集。源自 50 个真实的 28nm 芯片设计，涵盖从设计级到物理级的全流程数据	[10]
OpenROAD	持续更新	来自 OpenROAD Flow Scripts 仓库的开源 RISC-V 设计	[47,55]

2.2.2 与实际工业场景的数据差异

Mirhoseini 等人[3]和 Basso 等人[20]展示了真实工业场景的复杂性。他们分别使用了 Google TPU 的真实网表和英飞凌 130 nm 模拟电路数据，处理了包括时序违例、布线拥塞及器件匹配等实际指标。这些研究表明，只有经过海量、高复杂度数据验证的算法，才具备工业部署的潜力。但实际工业级芯片数据也面临着获取壁垒高、难以开源的问题，因此造成学术研究与业务落地之间的鸿沟。

2.2.3 合成数据的探索

为弥合数据鸿沟，Mallappa 等人[23]发布的 FloorSet 数据集及 Liu 等人[48]的合成数据生成策略具有重要意义。这些工作通过合成包含百万级模块且带有真实设计约束的数据，为学术界提供了更接近工业需求的标准化基准。

2.3 版图规划 AI 算法的演进与发展

回顾版图规划领域的 AI 研究，可以看到其方法路线和研究重点经历了较为清晰的演进过程。这一变化既反映了学术界对问题本质理解的加深，也体现了研究方向对工业需求的逐步回应。

2.3.1 从单一目标优化向多目标协同演进

早期的研究（如基于模拟退火和传统遗传算法的方法）主要聚焦于最小化芯片面积和线长这 2 个基础指标。然而，随着工艺节点的缩小，单一指标的优化已无法满足高性能芯片的需求。近年来，研究热点迅速转向多目标协同优化（multi-objective collaborative optimization）。例如，Mirhoseini 等人[3]和 Wang 等人[54]在奖励函数中明确引入了拥塞（congestion）、时序（timing）和密度（density）等多个维度的约束。Basso 等人[20]的工作将布线成功率直接纳入优化目标，标志着版图规划不再是孤立的优化问题，而是物理设计全流程优化的起点。

2.3.2 从纯几何拓扑向语义理解演进

在方法论上，传统的版图规划主要依赖 B*-Tree、序列对等纯几何拓扑表示进行搜索[14]。虽然这些方法在保证布局合法性方面表现优异，但往往忽略了电路网表本身蕴含的功能语义和结构特征。随着深度学习的引入，研究重心开始向“语义理解”转移。

GNN 和 HyperGCN 的广泛应用[3,46,53]表明,现在的算法试图先“读懂”电路图的连接逻辑和模块功能,再据此进行物理位置的推断。这种从“几何驱动”到“语义驱动”的转变,使得算法在处理未见过的电路设计时具备了更强的泛化能力。

2.3.3 强化学习方法的范式和架构创新

强化学习因其与序列决策问题的天然契合性,在版图规划的 AI 方法中占据主导地位。现有的强化学习方法在策略范式、状态感知及网络架构创新上呈现出多样化的演进趋势。

首先是构造型(constructive)与改进型(iterative)策略的演变。构造型策略从零开始逐个放置模块。Mirhoseini 等人[3]、Amini 等人[46]和 Wang 等人[54]将放置动作转化为拓扑序列的生成,训练策略网络直接输出宏单元位置,具有极高的推理速度。相比之下,改进型策略如 Xu 等人[52]和 Yu 等人[43]则从初始布局出发,通过交换、旋转等扰动动作迭代优化布局质量,在保证布局合法性和局部搜索能力上具有优势。

其次是 GNN 架构的改进。鉴于电路网表天然的图结构属性,GNN 已成为强化学习智能体感知状态的标准配置。Mirhoseini 等人[3]提出的 Edge-GNN 有效捕捉了节点与边的特征。Amini 等人[46]和 Guan 等人[53]则进一步引入超图卷积网络(HyperGCN),以更精准地描述一个线网连接多个引脚的真实电路连接关系。Wang 等人[54]设计的三头 GNN 架构(策略、价值、重建)则显著增强了特征提取的鲁棒性。

最后是混合架构与大模型技术的引入。为克服纯强化学习方法样本效率低、长序列决策困难等局限,混合架构应运而生。Li 等人[44]提出的“进化强化学习”框架 CORE,通过双向协同机制结合了强化学习的局部微调能力与进化算法的全局探索能力。Guan 等人[53]在 DQN 中引入 LSTM 层(DRQN),赋予智能体“记忆”能力以应对放置顺序的级联效应。最新的趋势如 Amin 等人[50]利用 Decision Transformer 架构,从离线数据中学习策略,将强化学习问题转化为序列建模问题,大幅提升了算法的可扩展性。

需要额外指出的是,Mirhoseini 等人[3]发表在《Nature》上的工作在学术传播和产业关注度上具有里程碑意义,它显著推动了强化学习在宏单元布局与版图规划中的研究热潮。然而,对该工作的结论外推范围仍应保持审慎理解。后续公开复现研究与学术评论的争议,主要集中在 4 个方面[56]:1)原始实验依赖专有设计数据与较复杂的工程流程,公开实现与论文描述之间长期存在细节缺口,因此可复现性受到持续质疑;2)比较基线的选择与评测设定是否足够公平,尤其是与经典启发式方法、现代解析式布局器及商业工具之间的对比,仍存在不同观点;3)该方法在实际效果上对预训练、聚类、后处理及较大计算资源的依赖程度较高,这使其“学习到的策略”与“工程技巧叠加”的边界并不总是清晰;4)宏单元布局阶段取得的优势能否稳定传导到完整工业流程中的最终质量(quality of results, QoR),仍有赖于后续标准单元布局、时钟树综合、布线与签核结果的系统验证。换言之,这一代表性工作的重要意义更多在于证明了学习型方法能够进入版图规划核心决策环节,并激发了后续大量研究,而不是意味着强化学习已经在所有场景下稳定优于现有 EDA 方法。对该类工作的合理评价,应同时考察其代理指标收益、可复现性、计算代价以及与完整物理设计流程的一致性。

2.4 版图规划 AI 算法与传统 EDA 工具的结合

版图规划处于芯片物理设计流程的起点,其结果不仅影响后续布局与布线的求解空间,也会受到后续阶段反馈的反向约束。因此,相关 AI 方法能否真正落地,很大程度上取决于其与现有 EDA 流程的融合深度。综合近年来的代表性研究,版图规划中的 AI 路线大致可分为点工具算法替代和与 EDA 流程深度集成 2 类。

2.4.1 点工具算法替代

部分学术研究[43-44,46,52-53]倾向于构建独立的版图规划器,重点优化线长和面积,却较少考虑与后续时序分析、电源完整性分析及布线工具的交互。这类方法在工业流程中往往难以直接部署:一方面,学术数据集与真实工业数据之间存在明显差距,且后者难以开源;另一方面,版图规划位于流程前段,其局部改进要转化为最终 QoR 提升,通常还需要与后续阶段进行复杂的联合优化。

2.4.2 深度集成与反馈闭环

部分学术研究选择了与现有 EDA 流程深度耦合。Basso 等人[20]在强化学习训练循环中集成了 A*布线引擎,构建了端到端的反馈闭环。Mirhoseini 等人[3]将布局结果送入力导向工具和时序分析器进行验证。Shi 等人[47]基于开源 EDA 流程 OpenROAD 开发,直接调用 OpenSTA 和布线工具优化时序指标。Liu 等人[48]则将其方法定位为“初始化工具”,利用 AI 快速生成初始解供传统 EDA 工具优化。这种“AI 辅助”与“全流程融合”的模式降低了工业界采纳新技术的门槛,也为将基于 AI 辅助的版图规划真正纳入 EDA

工具链提供了思路。

2.5 本阶段方法适用边界比较

版图规划中的 AI 方法当前更适合作为增强现有流程的决策组件，而非在所有设计场景下直接替代成熟 EDA 工具。需要特别警惕的是，公开基准上代理指标的改善未必能够稳定传导为完整物理设计流程中的签核 QoR 收益；同时，部分方法虽在推理阶段具有速度优势，但训练往往需要较大的算力投入、较长的收敛时间和复杂的工程管线。如果这些成本只能换来局部代理指标的改善，其综合价值就需要重新评估。未来更有前景的方向，是在保持流程可验证性的前提下，提升代理目标与最终 QoR 之间的一致性，并增强方法在真实工业约束下的迁移能力。

3 布局问题的 AI 求解算法

布局是超大规模集成电路（VLSI）物理设计流程中的核心环节之一，直接影响芯片的时序收敛性和可布线性，对芯片的 PPA 指标有重大影响。随着半导体工艺节点向 3 nm 及以下演进，布局问题面临着超大规模网表（百万级至亿级门）、复杂几何约束以及多物理场耦合的严峻挑战[14]。本节将系统综述 AI 方法在布局阶段的应用。

3.1 基于优化目标与应用场景分类

布局问题并非单一的几何优化任务，而是涵盖从宏观布局到微观合法化的多阶段、多目标系统问题。根据文献关注的优化目标和应用场景，可将现有研究划分为以下 6 个子领域。

3.1.1 拥塞与可布线性（routability）预测

在先进制程下，金属层资源紧张且设计规则复杂，布局阶段对线长（HPWL）的优化已不再等同于布线阶段的成功。布局与布线之间的鸿沟导致后期出现大量的拥塞（congestion）和设计规则违例（DRV）。针对这一问题，早期 Xie 等人[5] 较早提出 RouteNet，利用 CNN 进行可布线性预测。RouteNet 能够处理混合尺寸（mixed-size）设计，准确预测设计规则违例的热点分布。Wang 等人[57] 提出了晶格超图（LH-graph）建模方法，并在整个学习过程中保留网表数据，使拥塞信息能够在几何和拓扑上进行传播。在此表示基础上，进一步开发了异构图神经网络架构 LHNN，利用它进行精确的拥塞预测。Liu 等人[58] 提出了一种基于全卷积网络（FCN）的显式拥塞优化框架。该框架设计了一个包含 RUDY、PinRUDY 和 MacroRegion 三种特征通道的 FCN 模型来预测拥塞热图，并创新性地推导了拥塞成本关于单元位置的梯度，该方法在与 DREAMPlace[59] 结合中实现了显式的拥塞消除。Alhyari 等人[60] 则针对 FPGA 设计，提出了一个基于 CNN 的预测框架，利用线长密度（WLPA）和引脚密度等特征生成热力图，预测布局是否可布通，还能通过预测结果动态调整单元膨胀（cell inflation）系数。

3.1.2 可布线性驱动的布局优化

更深层次的探索来自于对布局和布线问题的统一数学建模。Chen 等人[24] 提出的 RUPlace 框架，利用交替方向乘子法（ADMM）将布局问题与全局布线问题耦合。该方法引入 Wasserstein 距离来正则化单元分布与布线需求的匹配，并通过双层优化（bilevel optimization）求解布线子问题，在多个工业基准上将拥塞降低为 1/6~1/4。Li 等人[61] 则在解析式布局中引入了可微分的全局拥塞函数（基于泊松方程）和基于动量的单元膨胀技术，同时动态优化引脚可访问性（pin accessibility），在保持线长竞争力的同时大幅减少了详细布线违例。

3.1.3 混合尺寸布局（mixed-size placement）

随着 SoC 设计复杂度的提升，芯片中集成了成百上千个不同尺寸的存储器和 IP 模块（宏单元），与数百万个标准单元混合放置，使得混合尺寸布局成为 NP-hard 难题。Cheng 和 Yan[62] 提出的 DeepPlace 采用了两阶段流程，首先利用强化学习智能体（结合 CNN 和 GNN 的多视图策略网络）放置宏单元，再使用 DREAMPlace 放置标准单元。Chiang 等人[63] 针对强化学习布局中顺序决策导致的信息不足问题，提出了迭代块移动策略和半并发移动机制，利用 DQN 生成高质量的布局原型。Lai 等人[41] 的 ChiPFormer 是基于离线强化学习的混合布局方法。它利用 Decision Transformer 架构[64]，从预先收集的离线布局数据中学习可迁移的策略，通过引入“电路令牌”（circuit token）来区分不同电路拓扑，ChiPFormer 实现了强大的零样本迁移能力。Hou 等人[65] 的 TransPlace 则构建了包含网表图和单元流（cell flow）的异构图神经网络，学习 SE(2) 不变的相对位置表示，实现了跨电路的知识迁移。

3.1.4 三维芯片与异构工艺节点的布局

随着摩尔定律放缓，3D 堆叠技术成为提升算力密度的关键，但也引入了跨层对齐、热耗散和异构工艺匹配等新挑战。Chen 等人[66] 针对采用混合键合（hybrid bonding）的异构 F2F 堆叠结构，提出了七阶段三维解析布局框架。该框架设计了多技术目标函数，同时优化线长和混合键合端子（HBT）成本，并通过

HBT 与标准单元的三维协同优化，解决了异构节点下模块形状变化和跨层对齐的难题。Hsiao 等人[67]提出的 DCO-3D 框架利用孪生 UNet 网络并发预测上下晶粒的拥塞图，并通过可微分的 GNN 单元扩散模型，在 X、Y、Z 轴上同时重新分布单元，有效缓解了 3D 堆叠带来的垂直拥塞问题。Zhao 等人[68]的 3D-Flow 则关注 3D 布局后的合法化问题，提出了基于网络流的全局合法化算法，允许单元在芯片间移动以最小化位移，解决了传统 2D 合法化在 3D 场景下的局限性。

3.1.5 时序驱动（timing-driven）的布局优化

时序收敛是高性能芯片设计的生命线，传统的时序驱动布局依赖于耗时的静态时序分析(STA)迭代，难以在优化循环中频繁调用，AI 方法的引入为时序驱动的布局带来了新的优化方向。

Guo 和 Lin[69]建立了一个完全可微分的时序驱动布局框架。他们将 STA 过程类比为深度神经网络的前向传播，实现了 Elmore 延时模型关于单元位置的梯度计算。这使得布局器能够直接优化全局时序指标(WNS 和总负时序裕量(TNS))，在 GPU 加速下实现了显著的分析加速和时序提升。He 等人[70]则提出了一种可微分的 DelayGBM 模型，通过前瞻性 RC 网络生成特征，并设计了引脚到引脚(pin-to-pin)的动态权重计算方法，在保证高精度时序预测的同时，实现了基于梯度的直接优化。

Du 等人[71]进一步打破了布局与门尺寸优化(gate sizing)的壁垒，提出了一个可微分的联合优化框架。通过将门尺寸建模为连续变量并使用线性插值计算延时梯度，该框架能在布局的同时调整门尺寸，相比传统分离流程将 TNS 改善了 77.1%。Ye 等人[72]则关注路由前的时序预测，提出了融合网表、布局和电源配送网络(PDN)多模态信息的预测框架，通过帕累托梯度融合算法解决了多目标(时序、IR drop、拥塞)优化的梯度冲突问题。

3.1.6 详细布局与合法化（legalization）

作为布局的最终阶段，详细布局需要在严格遵守设计规则的前提下，对标准单元位置进行合法化，并优化 PPA 指标[14]。在移动决策方面，Esmaeili 等人[73]探索了利用强化学习来指导详细布局过程。虽然该工作是针对 FPGA 场景，但其提出的基于 Tabular Q-Learning 和 Actor-Critic 的智能体模型，能够根据局部拥塞和线长奖励，智能选择单元的最佳移动策略，为 ASIC 详细布局的智能化提供了重要参考。Lim 和 Park[74]则尝试利用 GNN 进行无监督详细布局。他们将详细布局中的重叠消除、对齐约束和位移最小化建模为可微分的损失函数，利用 GNN 直接优化单元位置。这种方法打破了传统详细布局依赖顺序移动的局限，实现了基于梯度的全局微调。

3.2 数据基准与实际工业级应用场景差异

3.2.1 布局问题学术数据基准

当前布局问题学术数据依赖于相关比赛公开的工业级数据集，以及学术界开源的大规模、高质量数据集，如 CircuitNet[75]，其规模和特征丰富程度与实际工业场景的差距在逐渐缩小。一方面，为了缓解数据匮乏，Kahng 等人[76]提出了 PPA 感知的聚类方法，并通过机器学习加速的虚拟化布局布线框架生成大量高质量的训练样本。Mangalagiri 等人[77]提出的 CDLS 框架则利用生成式 AI 从历史模拟电路版图中提取设计约束，用于合成新的模拟布局数据。Lai 等人[41]在 ChiPFormer 中采用了电路令牌和多任务学习策略，实际上也是一种利用多样化电路数据增强模型泛化能力的手段。另一方面，ChiPBench[78]和 IEEE CEDA DATC[79]正致力于标准化 AI-EDA 数据的评价、生成与共享。学术界最近提出的 AiEDA 开源平台[10]也通过开源大规模芯片设计数据 iDATA，以及开源 AI 辅助设计库，为包含布局问题在内的芯片物理设计 AI 算法研究提供完整且标准化的数据管道解决方案。相关文献中使用的开源布局数据集如表 2 所示。

3.2.2 与实际工业场景的数据差异

然而正如 Wang 等人[78]在 ChiPBench 研究中指出的，现有的 AI 布局算法评估往往依赖于中间代理指标(如 HPWL)，这与最终的 PPA(如 WNS、TNS、功耗)存在严重脱节，缺乏包含复杂时钟树、多电压域和真实物理约束的工业级数据是制约 AI 模型泛化的瓶颈。实验显示，虽然某些算法在线长上表现优异，但在经过完整 EDA 流程后，其时序和功耗可能反而恶化。

3.3 布局 AI 算法的演进与发展

芯片布局问题的 AI 研究经历了从随机搜索到解析优化，再到如今与深度学习和 GPU 优化算法深度融合的演进过程。

3.3.1 从早期搜索到引入模型的解析式算法

以 TimberWolf[84]为代表的早期布局算法主要依赖随机搜索策略。该方法理论上可收敛至全局最优，但计算代价极高，随着电路规模增长至数十万门级，其运行时间变得不可接受。为应对规模挑战，Force-Directed[85]方法引入了力导向模型，通过模拟物理力场实现单元扩散。NTUplace[86]进一步完善了对数和

Table 2 Open-source Benchmarks for AI-driven Placement Algorithm

表 2 布局问题 AI 算法相关开源数据集

数据集	发布时间	数据集简介	使用该数据集的相关文献
ITC'99	1999	经典的测试电路数据集，布局规模较小	[70]
ISPD 2005	2005	混合尺寸布局基准测试集，包含大量标准单元和宏模块	[41, 62−63, 80]
ISPD 2015	2015	针对详细布线驱动的布局和拥塞消除的布局竞赛数据，包含复杂的宏模块阻塞和布线资源限制	[58, 61, 65, 80]
ICCAD 2015	2015	针对时序驱动布局的竞赛基准，数百万门级规模的设计	[65, 69]
CircuitNet 1.0/2.0	2022/2024	大规模开源 ML-EDA 数据集。基于商业 EDA 工具链生成，包含多模态布局数据，可开展时序任务支持	[24, 71, 75]
ChiPBench	2024	专门用于评估 AI 布局算法性能（PPA）的综合基准，旨在解决中间指标与最终 PPA 不一致的问题	[78]
iDATA	2025	基于 AiEDA 库生成的大规模结构化数据集。源自 50 个真实的 28nm 芯片设计，其中包含布局阶段全流程数据	[10]
OpenHW RISC-V	持续更新	现代高性能开源 RISC-V 处理器数据，可用于布线阶段算法设计和评估	[70]
OpenROAD	持续更新	源自 OpenROAD 项目维护的开源 RISC-V 设计集合	[76, 79]
OpenCores	持续更新	源自 OpenCores 开源社区的开源 IP 核，结合先进工艺节点进行物理实现以生成数据集	[81−83]

指数(log-sum-exp)平滑模型，显著提升了大规模电路的布局质量和速度。这一系列模型思想的引入为后续布局 AI 算法研究建立了坚实的基础。

3.3.2 从静电场模型到 AI 启发的 GPU 加速算法

ePlace[87] 和 RePlAce[88] 创新性地将布局密度约束建模为静电场系统，利用泊松方程求解电势能，极大地提升了全局布局的平滑性和收敛性。DREAMPlace[59] 及其后续改进算法受神经网络训练时的梯度计算启发，将这一物理模型的计算过程映射到深度学习框架(PyTorch)上，利用 GPU 的并行计算能力实现了超 30 倍的性能提升。在此基础上，将拥塞[58]、时序[69]等作为可微分项加入 DREAMPlace 损失函数，成为当前学术界最活跃的研究方向，布局研究重点已从单纯的线长优化转向 PPA 的全面协同优化。

3.3.3 从强化学习探索到 AI 深度赋能多目标优化

Google 的《Nature》论文[3]引发了基于强化学习的宏单元布局的研究热潮，但早期的在线强化学习方法布局效率低下，无法应用于较长的决策序列学习。近年来，研究趋势逐步转向离线强化学习(如 ChiPFormer[41])和混合优化策略(如 DeepPlace[62]、TransPlace[65])演进，以缓解样本效率和迁移性问题。另一方面，在网络结构选择上，GNN 因其对网表拓扑的强大表征能力，已成为连接逻辑综合与物理设计的关键桥梁。最新的研究，如 ParaFormer[83]、PTA-GNN[82]利用异构 GNN 和 Vision Transformer 来捕捉复杂的电路几何与拓扑特征，从而提升对寄生参数、时序和拥塞的预测与指导能力。

3.4 布局 AI 算法与传统 EDA 工具的结合

布局是芯片物理设计中的核心环节，其结果直接影响时序收敛性和可布线性，也深刻决定最终 PPA。一方面，由于布局问题具备较强的可微特性，部分 AI 驱动方法能够较自然地嵌入传统 EDA 流程[59]；另一方面，也有部分 AI 布局算法通过预测结果或约束生成，为传统 EDA 工具提供额外的优化指导信息。

3.4.1 点工具优化替代

早期的 AI 布局研究多表现为独立的点工具。例如，Szentimrey 等人[89]的 FPGA 拥塞预测器和 Alhyari 等人[60]以及 Xie 等人[5]的可布线性预测器。虽然它们能提供有价值的反馈，但由于缺乏与核心优化引擎的深层耦合，往往需要通过文件交换或脚本接口进行交互，效率受到 I/O 限制。

3.4.2 嵌入式 AI 指导工具

近年来，将 AI 算法嵌入传统 EDA 工具布局阶段提供额外指导信息，被认为是更具落地前景的路线。Cao 等人[81]和 Ding 等人[82]的工作并不直接进行布局，而是利用 GNN 预测门尺寸调整、缓冲器插入方案或生成区域约束(region constraints)。这些预测结果被转化为 Tcl 脚本，指导商业工具执行更优的详细布局。这种模式既利用了 AI 的全局预测能力，又保留了商业工具在 DRC 检查和合法化上的绝对优势。

3.4.3 深度集成的“白盒”优化

以 DREAMPlace[59]为代表的基于 AI 启发的开源布局框架正在推动 AI 布局算法与传统 EDA 流程的深度融合。DREAMPlace 不仅是一个布局器，更是一个可微分的优化底座。Liu 等人[58]、Guo 等人[69]、He 等人[70]的工作修改了 DREAMPlace 的底层梯度计算逻辑，将拥塞、时序等复杂目标“白盒化”为可微分算子。Liu 等人[80]的 Xplace 进一步通过算子级优化和傅里叶神经算子(FNO)增强，展示了深度学习模型如何作为核心组件内嵌于传统的解析式布局引擎

中,实现算子级别的协同进化。

3.5 本阶段方法比较与适用边界

从优化目标、输出形式和与EDA耦合深度看,现有AI布局方法大致可分为3类。第1类是预测与评估型方法,主要输出拥塞图、时序指标、可布线性评分或约束建议,适合作为早期分析器、参数调节器或脚本生成组件,工业接入成本较低,但通常不直接改变优化算法内核。第2类是可微分优化与白盒增强方法,它们通过修改解析式布局器的目标函数、梯度算子或求解流程,把拥塞、时序、IR drop等因素前移到布局阶段,因而与现有EDA流程耦合更深,也更容易把中间预测转化为实际QoR收益。第3类是面向混合尺寸布局、三维布局和离线强化学习迁移的直接决策方法,这类方法更强调直接输出布局方案或关键布局决策,在探索大搜索空间、处理新问题设定方面更具灵活性,但对训练数据覆盖范围、奖励设计和后续流程验证的依赖也更强。

就当前发展阶段而言,布局中的AI方法更适合沿着预测增强、白盒算子融合和局部决策自动化3条路线向工业落地推进,而不是在分散基准上单纯追求单项最优。需要持续警惕的是,许多工作报告的优势仍主要体现在HPWL、拥塞热图误差或预布局时序估计等代理指标上,这些收益能否在经历合法化、CTS、详细布线和后续ECO之后稳定转化为最终PPA改善,仍需在统一流程下验证。同时,对于依赖大规模训练、复杂模型堆叠或长序列强化学习的方案,其训练成本、复现实验门槛与跨设计泛化能力,也可能削弱工程吸引力。未来更关键的任务,是建立相同基准、相近流程和相近指标下的可比评测,并进一步提高代理目标、布局中间结果与最终签核QoR之间的一致性。

4 时钟树综合问题的AI求解算法

时钟树综合直接影响芯片的主频、功耗和鲁棒性。随着工艺节点演进至3 nm及以下,时钟网络在偏斜控制、功耗占比和物理规则复杂度等方面都面临更大压力[14]。传统基于规则和启发式的时钟树综合方法在处理这类多维、非凸优化问题时逐渐显得吃力,因此AI方法开始被引入这一阶段。

4.1 基于优化目标与应用场景分类

4.1.1 基于AI的时钟树综合结果预测

在时钟树综合早期(pre-CTS),准确预测时钟树结构以及综合后的QoR对于指导布局布线和减少迭代至关重要。Nagaria和Deb[90]提出了一种结合CNN预测与K-Means聚类的两阶段优化方法。首先利用CNN从布局图像中预测所需的缓冲器数量和线负载,然后利用K-Means算法对触发器进行聚类(flop-trays),并通过线性规划(LP)优化其位置以最小化位移和网络成本。Jain等人[91]针对传统时钟树综合优化耗时且需反复迭代的问题,提出了一种基于人工神经网络(ANN)的预测模型。该模型基于全连接层的架构,通过学习9个基准电路(涵盖TSMC 65 nm和90 nm工艺)的Pre-CTS特征,直接预测包含26个关键参数的Post-CTS QoR报告。Jain等人[92]进一步扩展了这一方法,旨在跳过耗时的时钟树综合和布线步骤。他们利用包含6个全连接层的ANN模型,直接从Pre-CTS阶段预测Post-Route的完整QoR数据。在TSMC 40 nm和65 nm工艺上的实验表明,该方法在几秒内即可提供高精度的预测反馈,大幅缩短了设计周期。

Du等人[93]提出了PowPrediCT框架,该框架采用创新的三阶段训练策略,基于预训练单元级GNN模型,然后利用布局特征训练专用的时钟线网功耗预测GNN,并且引入了注意力机制来识别寄存器簇,最后结合CNN提取布局空间特征进行校准,PowPrediCT将与商业工具Innovus之间的总功耗预测误差大幅降低,有效解决了早期功耗感知的“盲区”问题。

4.1.2 基于AI的时钟树参数辅助调优

Kahng等人[94]针对高维时钟树综合参数空间中的多重共线性问题,提出了分层混合代理建模(HHSM)方法。通过构建高精度的估计器,他们开发了SVM分类器来推荐最优的商业工具及其参数设置。Lu等人[95]则引入了GAN-CTS框架,利用条件生成对抗网络(cGAN)生成最优参数配置。该框架首先通过迁移学习(ResNet-50)从布局图像中提取设计特征,然后利用多任务回归模型高精度预测时钟树综合指标,最后通过cGAN的生成器生成能使指标最大化的参数集。

4.1.3 缓冲器插入与门尺寸优化

缓冲器插入(buffer insertion)和门尺寸优化是芯片时钟优化中最底层的物理操作。Du等人[96]提出了可微分的物理优化框架,将门尺寸调整和缓冲器插入统一在同一个数学模型中。该框架集成了时序感知的缓冲器树重构、可微分的延时/功耗模型以及离散感知的优化算法。利用GPU加速,该方法在ICCAD'24基准上比OpenROAD流程改善了23%的TNS,同时实现了30倍的速度提升。Wu等人[97]则利

用 GNN 和强化学习（PPO 算法）进行多目标优化。他们设计了专用的双向传播 GNN 架构来捕捉时钟网络中的信号流，并结合注意力机制预测缓冲器位置和类型。实验表明，该方法在复杂 SoC 设计中平均降低了 8.7% 的功耗和 6.3% 的最大偏差，且运行时间减少了 56.2%。

4.1.4 新型时钟树架构与拓扑生成

在探索时钟树架构方面，Ray 等人[98]开发了 TUNA 工具，利用贝叶斯优化算法自动探索多源时钟树（H 树结构）的最优参数组合，如最大扇出、目标偏差。该方法在保持时序收敛的同时，显著降低了时钟网络的功耗和 TNS。Choi 等人[99]则提出了一种基于接收端聚类和快速延迟预测的方法。该方法利用加权平衡 K-Means 算法优化 tap driver 的放置，并利用 CNN 模型快速评估不同驱动器数量下的最大子树延迟，从而高效搜索最优拓扑。Liu 等人[100]针对混合时钟树的顶层设计，提出了近似对称时钟树（approximate SCT）方法。该方法利用强连通分量（SCC）算法进行聚类，并利用 ANN 模型预测 U 型、Y 型和 H 型 3 种路由模式的延迟。

AI 技术也被用于探索双面时钟树等新型时钟网络。Bethur 等人[101]则利用无监督 GraphSAGE 模型对时钟端点进行聚类，智能化地选择部分关键路径布线到背面金属层。该方法在 3 nm 工艺下实现了 13% 的有效频率提升。Jiang 等人[102]提出了一个系统性的双面时钟树综合框架，结合分层 DME 布线和多目标动态规划，协同优化缓冲器和纳米硅通孔（nTSV）的插入。

4.2 数据基准与实际工业级应用场景差异

4.2.1 时钟树综合学术数据基准

多数时钟树综合 AI 算法研究基于 ISPD2009，2010 或 ISCAS’ 89 等公开基准电路，文献中所使用的开源数据集如表 3 所示。这些电路通常规模较小（数万门级），且缺乏现代工艺节点（如 FinFET）的物理约束（如多电压域、复杂设计规则）。例如，ISCAS’ 89 电路往往不包含复杂的时钟门控结构，这限制了模型在处理低功耗设计时的有效性。

4.2.2 与实际工业场景的数据差异

Jiang 等人[102]和 Ray 等人[98]虽然使用了部分工业级数据（如 TSMC 28nm/16nm），但由于 IP 保护，这些数据往往经过脱敏处理或无法公开，限制了模型的可复现性和泛化能力。Du 等人[96]使用了 ICCAD’ 24 竞赛的最新基准（基于 ASAP7 工艺），这在一定程度上缓解了数据陈旧的问题，但仍与真实的 3 nm/5 nm 工业设计存在差距。

时钟树综合的 AI 算法还面临着数据集间迁移困难的问题。时钟树综合结果对工艺节点（technology node）高度敏感。在一个工艺节点（如 65 nm）数据上训练的模型，往往难以直接迁移到另一个节点（如 5 nm）。目前的研究多通过迁移学习（如 GAN-CTS[95]）或重新训练来应对这一问题，但尚未形成通用的跨工艺解决方案。

4.3 时钟树综合 AI 算法的演进与发展

4.3.1 经典算法的奠基

早期的时钟树综合研究主要关注几何结构的对称性和线长平衡，为现代工具奠定了坚实的基础。MMM（method of means and medians）[105]通过递归划分终端集合并连接重心，实现了自顶向下的拓扑构建。GMA（geometric matching algorithm）[106]则采用自底向上的策略，通过最小权重匹配构建二叉树，显著减少了总线长。DME（deferred-merge embedding）算法[107]进一步引入了“延迟合并”策略，结合自底向上的拓扑构建和自顶向下的嵌入，在保证零偏差的同时最小化总线长，成为现代时钟树综合工具的核心算法

Table 3 Open-Source Benchmarks for AI-driven Clock Tree Synthesis Algorithm

表 3 时钟树综合 AI 算法相关开源数据集

数据集	发布时间	数据集简介	使用该数据集的相关文献
ISCAS’89	1989	经典的用于小规模电路测试的基准集	[92, 103−104]
ISPD 2010	2010	ISPD 2010 高性能时钟网络综合竞赛，包含来自商业公司的设计，具有大规模的时钟接收端和复杂的约束	[94, 100]
ISPD 2012	2012	源自 ISPD 2012 竞赛，关注逻辑门尺寸调整与时钟优化	[95]
ISPD 2013	2013	ISPD 2013 竞赛基准，关注评估可扩展的逻辑门尺寸调整和时钟优化	[92]
OpenPiton	2016	普林斯顿大学开发的开源多核处理器研究平台	[101]
CircuitNet	2022	大规模开源 ML-EDA 数据集。基于商业 EDA 工具链和 28 nm、14 nm 工艺生成，包含完整的布局布线信息	[96, 102]
ICCAD 2024	2024	ICCAD 2024 竞赛基准测试集，专注利用机器学习和 GPU 加速解决可扩展的逻辑门尺寸调整问题	[96]
OpenCores	持续更新	源自 OpenCores 开源社区的开源 IP 核，结合先进工艺节点进行物理实现以生成数据集	[91]

基础。

4.3.2 从传统算法到 AI 辅助时钟树参数优化

随着设计复杂度的增加，经典算法在处理多角多模(MCMM)和变异性时显得力不从心，AI 开始作为辅助工具介入。Kahng 等人[94]和 Nagaria 等人[90]利用机器学习模型预测时钟树综合结果或推荐工具参数，帮助设计者减少迭代次数。Lu 等人[95]引入对抗生成网络(GAN)生成最优参数配置，表明 AI 方法已能够从单纯预测进一步走向设计策略生成与主动探索。

4.3.3 从参数优化到拓扑生成和工艺协同

近年来，Liu 等人[100]和 Choi 等人[99]通过引入 AI 算法实现新型时钟树架构搜索与设计。Li 等人[108]提出的 SLLT 框架则通过构建 Skew-Latency-Load Tree，在经典拓扑结构上进行了创新，通过 CBS(concurrent BST-SALT)构建方法，实现了偏差、延迟和负载的协同控制，填补了现有 Steiner 树和偏差树之间的空白。面对 3 nm/5 nm 等先进工艺的具体挑战，相关研究开始深度结合物理底层，如针对背面时钟网络的研究[101-102]，显示了 AI 方法正在成为解决摩尔定律物理瓶颈的关键工具。

4.4 时钟树综合 AI 算法与传统 EDA 工具的结合

4.4.1 参数优化器与反馈预测

Kahng 等人[94]、Lu 等人[95]和 Ray 等人[98]通过引入机器学习和深度学习算法，分析历史数据推荐最优的工具参数，如 target skew 和 max fanout，指导商业工具(如 Innovus)运行。这种方式对现有 EDA 流程是非嵌入式的，它们不修改工具内部逻辑，而是优化工具的输入配置，实现对现有工具能力的挖掘。Jain 等人[92]利用包含 6 个全连接层的 ANN 模型，直接从 Pre-CTS 阶段预测 Post-Route 的完整 QoR 反馈数据，从而辅助传统 EDA 工具中的时钟树综合。

4.4.2 AI 算法嵌入传统 EDA 工具的时钟树综合

Wu 等人[97]和 Bethur 等人[101]将 AI 算法嵌入到传统 EDA 流程的特定环节，形成混合流程。例如利用 GNN 预测最优的缓冲器位置或时钟树拓扑，然后将这些预测转化为约束(constraints)，指导传统时钟树综合算法完成后续细节。这种模式结合了 AI 的数据学习能力和传统 EDA 工具在流程完整度和满足复杂约束的优势，实现了优势互补。

4.5 本阶段方法比较与适用边界

总体来看，时钟树综合阶段的 AI 方法相比版图规划和布局更依赖工艺条件与工具实现，研究重点也更集中于预测、参数调节、局部结构生成以及底层物理操作优化。从问题表示看，时钟树综合任务既可以视为布局驱动的空间分布问题，也可以视为由时钟接收端、缓冲器和线网组成的图结构问题，因此 CNN、ANN、GNN 和生成模型都能找到相应应用。其中，ANN 及传统机器学习方法更适合 QoR 预测、参数推荐和工具选择等低维映射问题，训练成本相对可控、部署门槛较低；GNN 与生成式模型则更适合缓冲器插入、拓扑生成和背面时钟网络等结构性更强的任务，但对训练数据质量、工艺特征完整性和约束建模精度要求更高。

就现阶段而言，时钟树综合中的 AI 方法更适合沿着预测反馈增强、参数推荐和局部物理优化 3 条路径稳步推进，而不宜被简单视为可独立替代成熟时钟树综合引擎的通用方案。需要特别关注的是，许多工作在公开数据上展示的收益主要体现为预测精度、局部功耗改善、偏斜下降或参数搜索效率提升，这些优势能否在真实工业流程中稳定传导为全芯片级的时序、功耗和鲁棒性改善，仍需与布局、布线、时序签核联合考察。同时，时钟树综合对工艺节点、寄生模型、时钟门控结构和工具内部实现高度敏感，即便某些方法在单一工艺和单一流程上效果突出，也未必能够直接迁移到其他节点或设计平台。未来更关键的方向，是提升跨工艺迁移能力，增强预测结果与签核 QoR 的一致性，并建立更贴近现代先进节点约束的公开评测体系。

5 布线问题的 AI 求解算法

布线是芯片后端物理设计中最复杂、也最耗时的环节之一，其核心任务是在满足设计规则检查(DRC)、时序和功耗等约束的前提下，将网表中所有线网(net)的逻辑连接转化为物理连线[14]。随着工艺节点演进至 3 nm 及以下，布线资源日益稀缺，设计规则呈指数级增长。传统的基于启发式算法的布线器在应对超大规模设计时，面临着收敛困难、运行时间过长以及存算墙瓶颈等严峻挑战，因此 AI 方法被广泛用于该阶段。

5.1 基于优化目标与布线阶段的分类归纳

布线过程通常被分为全局布线和详细布线两个阶段，不同阶段都涉及到底层线网拓扑生成或 Steiner 树构建[109]。根据布线 AI 算法研究相关文献聚焦的具体问题，可将其划分为 5 个领域。

5.1.1 AI 驱动的线网拓扑生成与 Steiner 树构建

线网拓扑生成与 Steiner 树构建是布线的底层核

心问题，直接影响线长和时序基础[110]。传统的Steiner最小树（SMT）构建依然是研究的热点，但近年来相关研究重点已转向处理更复杂的约束和多目标优化的Steiner树生成。

Utyamishev等人[111]将全局布线重新定义为图像到图像（image-to-image）的生成问题。该工作利用变分自编码器（VAE）和定制损失函数，直接从布局图像生成布线结果，并使用传统算法进行后处理。Liu等人[112]提出了REST框架，这是利用强化学习解决直角Steiner最小树（RSMT）构建问题的工作。不同于传统的启发式算法，REST将RSMT构建建模为序列决策问题，智能体通过学习边的选择策略来逐步构建树，在保证线长质量的同时，相比传统启发式算法具有更快的推理速度。针对生成式AI容易产生断线的问题，Du等人[7]提出了HubRouter两阶段框架：第1阶段利用深度生成模型预测“枢纽点（hubs）”，第2阶段利用actor-critic强化学习模型将这些枢纽点与引脚连接起来。这种先生成关键点，再保证连通性的策略，有效解决了AI模型在连通性上的缺陷，但仍然面临着拥塞较多的问题。

针对复杂约束和线网拥塞优化需求，Liu等人[6]提出NeuralSteiner算法。该方法采用了“预测—选择—连接”的三阶段框架：首先利用深度学习模型预测具备全局拥塞感知的Steiner点，再通过图算法筛选高质量候选点，最后完成连通性构建。相比前述AI方法，NeuralSteiner在显著降低拥塞的同时几乎不牺牲线长质量。Lin等人[8]提出EPST框架，旨在解决传统算法难以兼顾多种Steiner树结构的局限。该框架设计了一种通用的边点序列编码机制，并利用基于Transformer的编码器-解码器架构来学习引脚的空间关系。通过actor-critic强化学习训练，EPST能通过采样生成多样化的高质量拓扑，为后续布线拥塞优化提供丰富的候选方案。Chen等人[113]的RL Router则针对复杂的多层障碍物规避问题，结合了蒙特卡洛树搜索（MCTS）和3D U-Net。该方法利用MCTS直接探索Steiner点的组合空间，通过3D U-Net提取障碍物特征，从而在存在大量障碍物的场景下高效生成无冲突的SMT，显著提升了搜索效率和解的质量。

针对时序驱动的线网生成以及多目标权衡问题，HarRL[110]采用actor-critic机制，创新地引入谐波函数参数化来处理缓冲器位置的连续-离散混合动作空间，使得智能体能同时输出离散的拓扑连接和连续的缓冲器相位。Chen等人[114]的PatLabor利用强化学习策略迭代来训练局部搜索的评分函数，旨在帕累托前沿上寻找线长、延迟、偏差等多目标之间的最佳平衡点。Wang和Ding提出的CoTL[115]则结合了强化学习与核心集（coreset）技术，通过迁移学习快速适应新的多目标优化任务，有效解决了在不同设计目标间切换时重新训练模型的高昂成本问题。

由于线网拓扑生成与Steiner树构建是全局布线和详细布线共同依赖的底层问题，后续分类将重点讨论在全局布线与详细布线阶段差异更明显的AI方法。

5.1.2 全局布线中的AI算法

作为连接宏观布局与详细布线的桥梁，全局布线主要关注拥塞管理、资源分配和时序优化等方面。Fang等人[116]针对全局布线与详细布线资源模型不匹配导致后期DRC违例的问题，引入了容量缩减技术。该方法在全局布线阶段根据引脚密度和RUDY拥塞估计，动态降低特定区域（如宏单元周围）的边容量，为详细布线预留资源，该预处理技术能显著提升详细布线的收敛率。Liu等人[117]提出的CU全局布线引入了概率资源模型，在3D全局布线中引入概率机制来模拟详细布线的资源占用，从而生成更具鲁棒性的布线引导。Zeng等人[118]则提出在全局布线之前增加“网络资源分配”步骤，通过线性规划预先规划关键线网的资源分配，并建立全局布线与轨道分配之间的迭代反馈，有效缓解了拥塞热点。Hu等人[119]的GNN-MLS针对混合工艺节点3D IC，利用GNN辅助金属层共享。该方法构建了包含时序路径的图模型，通过GNN学习最优的层分配策略，以平衡层间资源利用率。

在时序驱动的全局布线方面，Liu等人提出的AiTPO[120]利用异构网络（KAN-UNet）在全局布线阶段进行精准的时序预测。该框架不仅预测时序违例，还据此主动进行线网重排序和重布线，将详细布线后的时序优化目标前移至全局布线阶段解决，显著改善了最终的WNS和TNS。Liu等人[121]提出的TSteiner则利用GNN构建签核（sign-off）时序评估器，建立了一个从后端精确时序分析到前端拓扑生成的反馈回路。通过梯度反向传播，该方法能直接优化布线拓扑以满足签核级时序要求。

面对超大规模设计（如ISPD 2024竞赛题），Zhao等人[122]提出的HeLEM-GR利用异构计算架构，将逻辑复杂的迷宫搜索和边缘计算卸载到GPU上，同时采用线性化指数乘子法（LEM）优化溢出。Lin等人[123]提出的InstantGR通过并行化A*搜索和批量并

发处理，实现了极高的布线速度。Zhang 等人[124]的 V-GR 则专注于通孔优化，在 GPU 加速的同时引入通孔感知的代价函数，有效减少了通孔数量和层间跳转。Li 等人[125]的 DGR 进一步提出了完全可微分的基于 GPU 加速的全局布线器，将离散的布线问题松弛为连续的概率优化问题，利用梯度下降并发求解所有线网的路径，展示了 AI 原生布线器的潜力。

5.1.3 详细布线中的 AI 算法

Li 和 Liu 等人[126]提出的 AiDRC 框架利用 AI 模型加速设计规则违例的预测和检查。它包含一个全局布线后的预测模型用于指导详细布线，以及一个详细布线迭代中的快速检查模型替代慢速的传统 DRC 引擎，显著提升了迭代效率。Zhao 等人[127]的 GTA 则针对轨道分配阶段，利用 GPU 加速和轻量级查找表实现冲突检测，解决了大规模并行轨道分配中的数据竞争和同步开销问题。Jiang 和 Fang[128]提出了并发详细布线与引脚图案再生技术，针对极端的引脚访问问题，允许布线器在布线过程中动态修改标准单元的引脚形状(pin pattern)，打破了传统库单元固定的限制，极大提升了引脚可访问性。

5.1.4 拥塞、可布线性与时序预测

Xie 等人[5]提出的 RouteNet 利用 CNN 进行可布线性预测，在不进行实际全局布线的情况下，准确预测设计规则违例的热点分布，证明了 CNN 在提取布局布线拥塞特征方面的强大能力。Zhou 等人[129]利用生成对抗网络(c-DCGAN)将布局视为图像生成拥塞热图，通过图像生成任务预测拥塞热点。Liu 等人[130]提出的 VALCE 通过变分推理增强了拥塞预测中的空间标签相关性，利用变分自编码器捕捉拥塞分布的潜在概率特性，提高了预测的鲁棒性。Lin 等人[131]提出的 URoute 则利用元学习框架，解决了跨工艺、跨设计的通用可布线性预测难题。该方法通过元训练学习不同设计风格的通用特征，使得模型能够快速适应从未见过的新设计或新工艺节点。

Zhang 等人[132]提出了基于迁移学习的时序预测框架。针对先进工艺节点数据稀缺的问题，该方法利用旧工艺节点的丰富数据进行预训练，并通过特征解耦与对齐技术将知识迁移到新工艺节点，实现了在小样本下的高精度时序预测。

5.1.5 特定领域的布线 AI 算法

Xu 等人提出的 AnalogFold[133]利用异构 3DGNN 指导模拟电路的性能驱动布线。该方法构建了包含器件、线网和物理约束的异构图，通过 3DGNN 捕捉模拟电路中复杂的电磁耦合效应，指导布线器避开敏感区域。Hsu 等人[134]的 TransRoute 探索了晶体管级的分层布线，试图替代传统的标准单元级流程，直接在晶体管层面进行连接，以获得极致的面积优化。

5.2 数据基准与实际工业级应用场景差异

5.2.1 布线问题学术数据基准

在研究所使用的数据集方面，除少部分研究仍在使用 ISPD98 等较小规模的布线数据，大部分研究(如 HeLEM-GR[122]、InstantGR[123]、DGR[125])使用的是 ISPD 2018/2019/2024 竞赛基准。这些数据源自工业界(如 Intel、Synopsys 等)，具有极高的真实性和挑战性。例如，ISPD 2024 的测试用例包含高达 6 000 万个线网，这是早期学术基准(如 ISPD 98/05)无法比拟的。这些基准通常包含真实的宏单元布局、复杂的电源地网络和详细的设计规则定义。另外，学术界高质量的开源数据集(如 CircuitNet[75])和开源研究平台(如 iEDA[11]，AiEDA[10]，OpenROAD[55] 等)也为布线问题的 AI 算法研究提供了广阔的空间，进一步弥合学术研究与实际工业场景的差距。相关文献所使用的开源数据集如表 4 所示。

部分涉及特定问题(如 PCB、模拟电路、3D IC)的研究使用了自建数据集或基于开源设计生成的合成数据。由于缺乏公开的模拟电路或 PCB 基准，研究者往往需要利用生成式 AI(如 Mangalagiri 等人[131])或程序化脚本来合成具有代表性的测试用例。

5.2.2 与实际工业级场景的数据差异

随着 ISPD 竞赛数据的普及，布线 AI 算法所处理的数据规模与真实工业场景之间的差距正在缩小，学术界已具备处理接近工业级规模(千万门级)问题的能力。HeLEM-GR 等人[122]工作明确展示了处理数千万线网的能力，这具有极高的工业参考价值。这表明学术界算法的扩展性已取得明显进展。

不过，尽管开源数据的规模不断扩大，学术数据集在工艺规则复杂度上仍存在明显简化，与真实工业需求仍有距离。比如性能较好的 DGR[125]和 RL Router[113]在简化的网格图上操作，可能无法覆盖先进制程(如 3 nm/5 nm)中极端复杂的物理约束，如多重曝光、复杂的过孔规则(via rules)、天线效应(antenna effect)等。

另一方面，在 Liu 等人[121]指出的一个关键痛点是，早期布线的优化目标与最终签核工具(如 Prime-Time, Innovus)给出的结果不一致[14]。虽然学术界试图通过建立 AI 模型(如 AiTPO[120])来拟合签核工具，但在真实工业流程中，设计往往仍依赖签核工具反

Table 4 Open-source Benchmarks for AI-driven Routing Algorithm

表 4 布线 AI 算法相关开源数据集

数据集	发布时间	数据集简介	使用该数据集的相关文献
ISCAS89	1989	经典的用于小规模电路测试的基准集	[126]
ISPD98	1998	ISPD 1998 竞赛发布的布线基准	[6−7]
ISPD 2011	2011	ISPD 2011 竞赛发布的基准，主要用于评估物理设计中的布线拥塞评估和预测技术	[130]
ICCAD 2015	2015	ICCAD 2015 时序驱动的布局竞赛发布的基准，也用于评估时序驱动的布线算法	[114−115, 131, 135]
ICCAD 2019	2019	ICCAD 2019 竞赛发布的全局布线基准，包含大规模的工业级设计用例，重点评估三维空间中的布线能力	[116−117, 124−125]
ISPD 2018/2019	2018/2019	ISPD 2018/2019 竞赛发布的详细布线基准，包含复杂的现代工艺设计规则，用于评估详细布线器的质量	[6, 118, 125, 127−128, 136]
CircuitNet 1.0/2.0	2022/2024	大规模开源 ML-EDA 数据集。基于商业 EDA 工具链和 28 nm/14 nm 工艺生成，包含完整的布线信息	[75, 131]
ISPD 2024	2024	ISPD 2024 竞赛发布的大规模全局布线基准，包含高达 5 000 万单元和 6 000 万线网的超大规模设计，评估算法的可扩展性和 GPU/ML 加速技术的应用	[122−123]
iDATA	2025	基于 AiEDA 库生成的大规模结构化数据集。源自 50 个真实的 28 nm 芯片设计，涵盖布线全流程数据	[10, 118, 126]
OpenROAD	持续更新	源自 OpenROAD 项目维护的开源 RISC-V 设计集合	[118, 126]

馈和 ECO 流程不断修正。由于高性能签核工具极其复杂且闭源，二者实现完全一致仍然十分困难。

5.3 布线问题 AI 算法的演进与发展

随着工艺演进和 AI 技术发展，布线 AI 研究的方法路线与关注重点呈现出较清晰的演进脉络。

5.3.1 从单一指标预测到 AI 原生布线生成

早期的布线问题的 AI 研究（2019—2022 年）多集中在拥塞预测、可布线性预测等较为单一的辅助任务，AI 主要提供热图等辅助信息。而最近的研究（如 EPST[8], HarRL[110], DGR[125]）更多转向生成式任务，即由 AI 模型或 AI 启发的 GPU 加速算法直接输出布线拓扑或完整方案，这标志着 AI 开始深入布线算法的核心逻辑。

5.3.2 从二维简化到真实三维场景与多物理场

早期的全局布线常将三维空间投影为二维处理，忽略了层分配的复杂性。近年来相关工作（如 CUGR[117,124], DLU[137]）进一步强调直接三维布线，直接处理层分配和过孔优化，以应对多层金属互连的挑战。此外，研究维度也在扩展，从单纯的线长/拥塞，扩展到时序优化、电源完整性和热/应力等多物理场考量，反映了物理效应在先进工艺中的主导地位。

5.3.3 从使用 AI 辅助到 AI 与异构计算加速结合

面对制程升级与设计规模扩张带来的双重压力，单纯依靠 AI 优化点工具已不足以支撑超大规模布线问题。GPU 加速（如 HeLEM-GR[122]、InstantGR[123]）已成为解决超大规模布线问题的标配手段，DGR[125] 工作更是结合了 AI 启发的可微分建模。预计未来该领域将探索算法智能化与算力优化相结合的协同演进路线。

5.4 布线问题 AI 算法与传统 EDA 工具的结合

5.4.1 孤立的预测与辅助工具

这类 AI 布线方法不直接修改设计数据，而是作为预测器或判别器，在某个阶段（通常为全局布线前或全局布线后）运行，为后续布线步骤提供指导信息。

Zhou 等人[129] 的拥塞预测算法，以及后续的 VALCE 算法[130] 和 URoute 算法[131] 作为全局布线之前的分析步骤，输入布局信息，输出拥塞或可布线性热图。这些热图可以帮助物理设计工程师手动调整布局，或者作为参数输入给后续的布线器。AiDRC[126] 包含全局布线后的预测模型和详细布线迭代中的快速检查模型，与 EDA 工具结合较紧密，但仍属于流程辅助加速组件，用于快速过滤明显的 DRC 违例，减少调用昂贵的签核级 DRC 工具的次数。Zhang 等人[132] 的时序预测主要用于早期设计探索或 PPA 评估，属于辅助决策工具。这些工具通常通过文件交换或 API 调用与主流程交互，集成成本低但反馈回路较长。

5.4.2 与传统 EDA 紧耦合的内核增强与替代

此类研究提出的方法能够直接替换或显著增强现有 EDA 布线流程中的核心引擎，具有较强的“即插即用”潜力。

Qi 等人[136] 提出的 IPAG 作为一个高性能详细布线器，直接替代现有的详细布线引擎。IPAG 内部集成了自适应的全局布线引导机制，能够直接读取标准的 DEF 文件和 LEF 库，输出最终的布线结果。DGR[125] 提出了完全可微分的并行全局布线器，用作 AI 启发的基于梯度的优化框架重构传统的全局布线算法核心。GTA[127] 专门针对连接全局布线和详细布

线的“轨道分配”环节进行 GPU 加速。这些模块完全嵌入在标准流程中,接收布局结果,输出布线引导(route guides),其输入输出格式与现有工业标准兼容。

除此之外,在底部算子层面,EPST[8]、RL Router[113]、HarRL[110] 和 NeuralSteiner[6] 等 Steiner 树生成算法,通常可以作为“库函数”被集成在更大型的布线器内部,解决局部的连接问题。

5.4.3 AI 驱动的多阶段联合优化

这类结合方式旨在借助 AI 打破 EDA 流程各步骤之间的割裂状态,通过引入后段设计步骤的预测或生成信息,支持跨阶段协同决策,从而实现更全局的优化。

Liu 等人提出的 TSteiner[121] 建立了一个从签核时序分析到早期布线拓扑生成的反馈回路,利用梯度反向传播直接优化拓扑。这种反馈机制使得早期的树综合能够感知后端的时序约束。Jiang 等人[128] 在详细布线阶段根据需求动态修改标准单元的引脚形状,打破了“布局”与“布线”以及“库设计”之间的界限,实现了单元库与布线的协同优化。AiTPO[120] 在全局布线阶段利用异构网络预测详细布线后的时序,并主动进行优化,将详细布线阶段的问题前移解决。这种结合方式能显著减少设计迭代,是未来 EDA 智能化非常具有前景的方向。

5.5 本阶段方法比较与适用边界

总体来看,布线阶段的 AI 方法是 4 个物理设计阶段中最贴近几何规则、制造约束和签核反馈的一类,因此其问题表示、优化目标与工程落地性之间的耦合最为紧密。从问题表示看,现有方法大致有 3 类路线:第 1 类路线将布局与布线状态表示为图像、热图或网格张量,适合拥塞预测、可布线性分析和早期时序评估,便于利用成熟的 CNN 或生成模型进行快速推理;第 2 类路线将线网、引脚、障碍物和层间关系表示为图或超图,更适合 Steiner 树生成、层分配和复杂拓扑搜索;第 3 类路线直接在可微分或并行搜索框架中重构布线内核,将传统离散搜索问题转化为更适合 GPU 加速或梯度优化的求解过程。布线方法对物理规则细节和状态空间爆炸更加敏感,因此任何表示上的简化都可能直接影响工业有效性。

若从学习范式、输出形式和与 EDA 的耦合深度来看,现有方法大体可分为 4 类。第 1 类是预测与评估型方法,主要输出拥塞热图、可布线性评分、DRC 风险或时序估计,适合作为流程前端的快速分析器,部署较轻量,但通常只改善决策信息而不直接修改布线解。第 2 类是拓扑生成与 Steiner 树构建方法,它们直接参与局部连接结构的生成,在降低线长、拥塞或延迟方面具有较强潜力,但通常仍需依赖后处理来保证连通性、规则满足和全局一致性。第 3 类是面向全局布线或详细布线内核增强的算法,例如可微分全局布线、GPU 加速轨道分配和高性能详细布线引擎,这类方法最接近工业核心流程;若输入输出能够兼容现有标准,其落地价值最高,但对工程实现复杂度和规则覆盖度要求也最苛刻。第 4 类是跨阶段联合优化方法,它们尝试将签核时序、引脚可达性或详细布线反馈前移到全局布线或拓扑生成阶段,在方法论上最契合现代先进工艺的闭环需求,但系统耦合深、验证成本高。

现阶段,布线中的 AI 方法更适合沿着预测增强、局部拓扑生成、内核级加速和跨阶段反馈优化 4 条路线逐步推进,而不宜在简化规则或单一指标上取得优势后就宣称具备全面替代传统布线器的能力。需要特别重视的是,公开基准上关于拥塞、线长或局部 DRC 的改善,并不天然等同于最终签核 QoR 的提升;在先进节点下,过孔规则、天线效应、多重曝光、引脚可达性和 ECO 修复等因素都可能改变早期优化结论。同时,若某些方法需要复杂训练流程、大量采样或专门的异构硬件支持,其综合收益还必须与工程接入成本一并评估。未来更关键的方向,是在保持规则完备性和签核一致性的前提下,提高方法在真实工业约束下的扩展性、可验证性和跨设计迁移能力。

6 总结与展望

随着集成电路工艺节点持续迈向 3 nm 及以下,芯片物理设计在计算复杂度、多目标优化和设计收敛等方面承受着越来越大的压力。AI,尤其是深度学习和强化学习等方法,为这一领域带来了数据驱动的优化能力,并推动设计范式由经验驱动逐步转向学习驱动。总体而言,现阶段 AI 对芯片物理设计的贡献主要体现在 4 个方面:其一,通过数据学习与模式识别提升预测与评估精度;其二,借助梯度优化与并行计算加速大规模布局布线求解;其三,通过端到端和跨阶段学习机制推动全流程协同优化;其四,为先进工艺下的 IR drop、电迁移和热耦合等复杂物理效应提供新的建模与优化思路。这些进展不仅推动了学术研究深入发展,也为工业界提升设计效率和 PPA 表现提供了技术储备。

然而，尽管 AI 在芯片物理设计中展现出巨大潜力，其实际落地仍面临多重局限与挑战：

1）高质量数据稀缺。由于高性能商业工具和芯片设计具有高度的知识产权与隐私保护属性，高质量、大规模、标注完整的工业级数据集稀缺。学术研究多依赖早期数据基准或简化后的竞赛数据，其规模、复杂度与真实物理设计场景存在显著差距，导致模型泛化能力不足，制约了学术成果向产业落地的转化。此外，由于 AI 模型对工艺节点敏感性强，跨节点迁移学习尚未形成通用解决方案。

2）模型的可解释性与鲁棒性不足。现有 AI 方法在可解释性、鲁棒性和设计规则合规性方面仍有短板，许多研究呈现明显的“黑盒”优化特征，难以通过显式规则或约束解释模型的执行逻辑与结果，也难以为工程师提供符合设计直觉的决策依据。与此同时，不少方法仍主要围绕代理指标（如 HPWL）进行训练和评估，与最终签核指标（如 WNS、功耗、良率）之间存在明显鸿沟，难以直接映射到制造层面的真实物理约束[11]。

3）与 EDA 流程集成深度不足。AI 工具与传统 EDA 流程的融合尚处于早期阶段。多数研究仍以点工具形式存在，缺乏与后端物理设计布局、布线、时序分析等环节形成深度闭环的能力。工业流程中复杂的数据格式、接口协议与工具链依赖，使 AI 模型嵌入成本高、迭代周期长。此外，AI 方法在计算资源消耗、训练时间与推理效率方面仍需进一步优化，以满足实际项目对时间与成本的严苛要求。

展望未来，AI 与传统 EDA 物理设计工具的融合将朝着更深层次、更紧耦合、更自动化的方向演进，并可能催生新一代 AI 原生 EDA 平台。结合当前学术界与产业界的探索进一步细化，本文认为后续值得重点关注的高价值方向主要包括以下 5 类。

6.1 电路大模型的构建与应用

电路大模型（large circuit model, LCM）代表了将预训练范式系统引入芯片物理设计的重要方向。其核心目标是利用大规模网表、布局、布线、时序、功耗、DRC 报告以及工程日志等异构数据，学习统一、可迁移、可复用的电路表示，从而为拥塞预测、布局初始化、时序评估、跨设计迁移和设计知识检索等任务提供通用基础模型。与当前针对单一任务训练专用模型的路线相比，电路大模型更有潜力缓解数据碎片化和任务割裂问题，并为后续的多任务协同优化提供共享表示基础。

然而，电路大模型要真正落地，首先需要解决多模态电路数据的统一表示问题，如图结构网表、二维或三维版图、时序路径、工艺规则和文本化设计报告等不同模态之间明显的语义差异和尺度差异。其次，工业级高质量预训练数据难以公开，导致模型规模、覆盖范围和泛化能力受限的问题也亟待解决。

6.2 AI 原生 EDA 的范式演进

AI 原生 EDA 体现的是比 AI 增强传统工具更进一步的范式转变，即从底层数据表示、优化目标到求解内核，都围绕 AI 可学习性与可微性重新组织设计流程。在这一方向下，AI 不再只是传统工具链中的附加模块，而可能直接参与布局、布线、时序优化乃至多目标权衡的核心求解过程，并与 GPU、异构加速器和云计算基础设施深度耦合，形成新的算法-算力协同框架。

这一方向的关键难点在于，传统 EDA 的很多核心步骤本质上是强约束、离散化、规则密集型的组合优化问题，难以直接转化为稳定、可训练、可验证的 AI 求解过程。即便可通过可微近似、代理目标或神经启发式方法重构部分内核，也常常面临优化目标与最终签核结果不一致的问题。此外，AI 原生 EDA 若要进入真实工业流程，还必须满足规则完备性、结果可复现性、运行效率和接口兼容性等工程要求。

6.3 智能体化设计流程与闭环构建

随着大语言模型和工具调用框架的发展，芯片物理设计中的 AI 角色正由单点优化器扩展为设计助手、流程协调器以及面向局部闭环的多智能体系统。更具吸引力的方向，是让智能体在受控工具接口下完成脚本生成、参数搜索、异常定位、设计知识检索、跨阶段结果分析和多模块协同优化等任务，并在人类工程师监督下形成“分析—决策—执行—反馈”的闭环流程。这种模式的价值不仅在于自动化重复劳动，更在于把分散在工具、脚本和工程经验中的隐性知识组织成可复用的流程能力。

但智能体化流程同样面临突出难点。首先，物理设计流程依赖大量商业工具和私有接口，智能体受制于接口开放度和环境可控性。其次，物理设计中的许多决策具有长时程依赖，一个局部参数选择可能在后续布局、布线、时序收敛甚至 ECO 阶段才暴露问题，这使得智能体面临远比一般软件自动化更复杂的延迟反馈问题。最后，若缺乏可追溯的决策记录、权限隔离与签核级验证机制，智能体即使能够提升局部效率，也难以被工业流程真正信任和大规模采用。

6.4 工业级数据基础设施与基准生态

从更基础的层面看，未来 AI-EDA 的发展很大程度上取决于工业级数据基础设施的建设。无论是训练电路大模型、评估跨设计泛化能力，还是支持闭环优化系统持续学习，都需要覆盖多工艺节点、多设计风格、多物理效应和多阶段流程的高质量数据资产。因此，建立标准化的数据格式、特征抽取接口、流程日志规范、任务定义与评测基准，将推动学术研究走向工程可复现和产业可迁移。若与开源芯片社区、云原生平台和高性能计算环境结合，还可能进一步推动共享工具链、远程训练和大规模协同实验的发展。

这一方向最突出的难点是数据的可获得性、可共享性和可比性。工业设计数据通常涉及知识产权、商业机密和工艺保密，难以形成真正开放的大规模数据集，导致模型训练和横向比较缺乏统一基础。此外，当前很多公开基准仍偏向简化场景，难以充分覆盖先进工艺中的复杂规则、真实约束。因此，未来不仅需要更多数据，更需要与真实工业场景严格对齐的数据治理、评测协议和反馈闭环。

6.5 可信 AI-EDA 与可验证部署

随着 AI 方法逐渐接近物理设计核心环节，可信 AI-EDA 将成为决定其能否进入工业主流程的关键研究方向。这不仅包括常见的可解释性与鲁棒性，还包括结果是否满足物理约束、是否与签核工具一致、是否能够在不同设计与工艺节点上稳定迁移，以及当模型出错时是否具备可诊断、可回退机制。对于芯片物理设计这一高成本、高风险场景而言，可信性是 AI 系统从研究原型走向工业工具的基本门槛。

该方向的核心难点在于，现有许多 AI 方法在面对分布偏移、极端约束或未见过的设计模式时可能出现不稳定行为，而这种不稳定在物理设计中往往具有放大效应。为此，未来研究需要探索将物理规则、专家先验、形式化约束与不确定性估计显式融入模型与流程之中，并建立覆盖训练、推理、验证和回退的保障机制。只有当 AI 方法能够在性能、透明度与可验证性之间取得平衡时，其在 EDA 中的大规模部署才具备现实基础。

总体而言，上述 5 个方向并非彼此孤立，而是共同构成未来 AI-EDA 演进的关键支柱。电路大模型提供统一表示基础，AI 原生 EDA 推动优化范式重构，智能体化流程提升系统级协同能力，工业级数据基础设施提供持续学习与公平评测的支撑，而可信 AI-EDA 则为这些能力进入工业场景提供必要保障。随着这些方向逐步成熟，芯片物理设计有望从当前的局部智能增强进一步迈向数据驱动、闭环优化、可信部署的新阶段。

作者贡献声明：刘睿之、李捷鹏完成相关文献资料的收集和分析，是论文的主要撰写人；王睿、隋京言参与文献资料的分析和整理，并对论文进行修改；李兴权、解壁伟、于春功、卜东波对论文选题、论文结构和文献收集提出指导意见，并对论文进行修改。

参 考 文 献

[1] Huang Guyue, Hu Jingbo, He Yifan, et al. Machine learning for electronic design automation: A survey[J]. ACM Transactions on Design Automation of Electronic Systems, 2021, 26(5): 1−46

[2] Kahng A. Machine learning applications in physical design: Recent results and directions[C]//Proc of the 2018 Int Symp on Physical Design. New York: ACM, 2018: 68−73

[3] Mirhoseini A, Goldie A, Yazgan M, et al. A graph placement methodology for fast chip design[J]. Nature, 2021, 594(7862): 207−212

[4] Guven I, Parlak M, Lederer D, et al. AI-driven integrated circuit design: A survey of techniques, challenges, and opportunities[J]. IEEE Access, 2025, 13(1): 167364−167389

[5] Xie Zhiyao, Huang Yuhung, Fang Guanqi, et al. RouteNet: Routability prediction for mixed-size designs using convolutional neural network[C]//Proc of the Int Conf on Computer-Aided Design. New York: ACM, 2018: 1−8

[6] Liu Ruizhi, Zeng Zhisheng, Ding Shizhe, et al. NeuralSteiner: Learning Steiner tree for overflow-avoiding global routing in chip design[C]//Proc of the 38th Int Conf on Neural Information Processing Systems: Vol. 37. Red Hook: Curran Associates Inc., 2024: 127346−127368

[7] Du Xingbo, Wang Chonghua, Zhong Ruizhe, et al. HubRouter: Learning global routing via hub generation and pin-hub connection[C]//Proc of the 37th Int Conf on Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2023: 78558−78579

[8] Lin Zhenkun, Liu Genggeng, Huang Xing, et al. A unified deep reinforcement learning approach for constructing rectilinear and octilinear steiner minimum tree[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025, 44(7): 2711−2724

[9] Attaoui Y, Chentouf M, Alaoui Ismaili Z, et al. Machine learning in VLSI design: A comprehensive review[J]. Journal of Integrated Circuits and Systems, 2024, 19(2): 1−14

[10] Qiu Yihang, Huang Zengrong, Tao Simin, et al. AiEDA: An open-source AI-aided design library for design-to-vector[J]. arXiv preprint, arXiv. 2511.05823, 2025

[11] Li Xingquan, Huang Zengrong, Tao Simin, et al. iEDA: An open-

source infrastructure of EDA[C]//Proc of the 29th Asia and South Pacific Design Automation Conf. Piscataway, NJ: IEEE, 2024: 77−82

[12] Bao Yungang. Analysis of the impact of artificial intelligence on technological development in the chip design field[J]. Microelectronics & Computer, 2025, 42(10): 1−8
(包云岗. 关于人工智能对芯片设计领域技术发展的影响分析[J]. 微电子学与计算机, 2025, 42(10): 1−8)

[13] Chen Lei, Chen Yiqi, Chu Zhufei, et al. Large circuit models: Opportunities and challenges[J]. Science China (Information Sciences), 2024, 67(10): 200402

[14] Kahng A, Lienig J, Markov I, et al. VLSI Physical Design: From Graph Partitioning to Timing Closure[M]. Cham: Springer, 2022

[15] Berndt A, Fogaça M, Meinhardt C. A review of machine learning in logic synthesis[J]. Journal of Integrated Circuits and Systems, 2022, 17(3): 1−12

[16] Fang Wenji, Wang Jing, Lu Yao, et al. A survey of circuit foundation model: Foundation AI models for VLSI circuit design and EDA[J]. arXiv preprint, arXiv: 2504.03711, 2025

[17] Yan Junchi, Lyu Xianglong, Cheng Ruoyu, et al. Towards machine learning for placement and routing in chip design: A methodological overview[J]. arXiv preprint, arXiv: 2202.13564

[18] Li Lin, Cai Yici, Zhou Qiang. A survey on machine learning-based routing for VLSI physical design[J]. Integration, 2022, 86(1): 51−56

[19] Zhou Zhanwen, Zhuo Hankui. Survey on intelligent routing approaches for chips[J]. Acta Automatica Sinica, 2024, 50(9): 1671−1703 (in Chinese)
(周展文, 卓汉逵. 芯片智能布线方法研究综述[J]. 自动化学报, 2024, 50(9): 1671−1703)

[20] Basso D, Bortolussi G, Videnovic-Misic M, et al. Advancing routing-awareness in analog ICs floorplanning[J]. arXiv preprint, arXiv: 2510.15387, 2025

[21] Wei Qingye. A review on meta-heuristic and reinforcement learning for VLSI floor planning[J]. International Core Journal of Engineering, 2025, 11(6): 1−23

[22] Lopera D, Servadei L, Kiprit G, et al. A survey of graph neural networks for electronic design automation[C]//Proc of the ACM/IEEE 3rd Workshop on Machine Learning for CAD (MLCAD). Piscataway, NJ: IEEE, 2021: 243−249

[23] Mallappa U, Mostafa H, Galkin M, et al. FloorSet: A VLSI floorplanning dataset with design constraints of real-world SoCs[J]. arXiv preprint, arXiv: 2405.05480, 2024

[24] Chen Yifan, Mai Jing, Zhang Zuodong, et al. RUPlace: Optimizing routability via unified placement and routing formulation[C]//Proc of the 62nd ACM/IEEE Design Automation Conf. New York: ACM, 2025: 1−7

[25] Cox D. The regression analysis of binary sequences[J]. Journal of the Royal Statistical Society, 1958, 20(2): 215−232

[26] Cortes C, Vapnik V. Support-vector networks[J]. Machine Learning, 1995, 20(3): 273−297

[27] Ahmed M, Seraj R, Islam S. The k-means Algorithm: A comprehensive survey and performance evaluation[J]. Electronics, 2020, 9(8): 1295−1314

[28] Shaikhina T, Lowe D, Daga S, et al. Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation[J]. Biomedical Signal Processing and Control, 2019, 52(1): 456−462

[29] Frank E, Trigg L, Holmes G, et al. Technical note: Naive Bayes for regression[J]. Machine Learning, 2000, 41(1): 5−25

[30] Chen Tianqi, Guestrin C. XGBoost: A scalable tree boosting system[C]//Proc of the 22nd ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2016: 785−794

[31] Ke Guolin, Meng Qi, Finley T, et al. LightGBM: A highly efficient gradient boosting decision tree[C]//Advances in Neural Information Processing Systems: Vol. 30. Red Hook: Curran Associates, Inc., 2017: 1−9

[32] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436−444

[33] Rumelhart D, Hinton G, Williams R. Learning Representations by Back-Propagating Errors[M]//Neurocomputing: Foundations of Research. Cambridge, MA: MIT Press, 1988: 696−699

[34] Hung Wei Tse, Huang Jun Yang, Chou Yih Chih, et al. Transforming global routing report into DRC violation map with convolutional neural network[C]//Proc of the 2020 Int Symp on Physical Design. New York: ACM, 2020: 57−64

[35] O'Shea K, Nash R. An introduction to convolutional neural networks[J]. arXiv preprint, arXiv: 1511.08458, 2015

[36] Chen Jingyi, Lv Zheqi. Graph neural networks for critical path prediction and optimization in high-performance ASIC design: A ML-driven physical implementation approach[C]//Proc of the Global Conf in Advanced Science and Technology: Vol. 1. 2025: 23−30. [2025-10-21]. https://proceedings.gcast.org/index.php/pr/article/view/4

[37] Ji H, Kim J, Choi J, et al. LayNet: Layout size prediction for memory design using graph neural networks in early design stage[C]//Proc of the 29th Asia and South Pacific Design Automation Conf. Piscataway, NJ: IEEE, 2024: 484−490

[38] Niu Zhaoyang, Zhong Guoqiang, Yu Hui. A review on the attention mechanism of deep learning[J]. Neurocomputing, 2021, 452(1): 48−62

[39] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint, arXiv. 1409.0473, 2014

[40] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems: Vol. 30. Red Hook: Curran Associates, Inc., 2017: 1−13

[41] Lai Yao, Liu Jinxin, Tang Zhentao, et al. ChiPFormer: Transferable chip placement via offline decision transformer[J]. arXiv preprint, arXiv. 2306.14744, 2023

[42] OpenAI, Achiam J, Adler S, et al. GPT-4 technical report[J]. arXiv preprint, arXiv. 2303.08774, 2023

[43] Yu Shenglu, Du Shimin, Yang Chang. A deep reinforcement learning floorplanning algorithm based on sequence pairs[J]. Applied Sciences, 2024, 14(7): 2905

[44] Li Pengyi, Kai Shixiong, Hao Jianye, et al. CORE: Collaborative optimization with reinforcement learning and evolutionary algorithm

for floorplanning[C]//Proc of the 39th Annual Conf on Neural Information Processing Systems. New York: Curran Associates, Inc., 2025: 1−9

[45] Li Zirui, Tian Kanglin, Zhai Jianwang, et al. FTAFP: A feedthrough-aware Floorplanner for hierarchical design of large-scale SoCs[C]// Proc of the 30th Asia and South Pacific Design Automation Conf. New York: ACM, 2025: 886−892

[46] Amini M, Zhang Zhanguang, Penmetsa S, et al. Generalizable floorplanner through corner block list representation and hypergraph embedding[C]//Proc of the 28th ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2022: 2692−2702

[47] Shi Yunqi, Lin Xi, Xu Siyuan, et al. ReMaP: Macro placement by recursively prototyping and periphery-guided relocating[C]//Proc of the 62nd ACM/IEEE Design Automation Conf. New York: ACM, 2025: 1−6

[48] Liu Yiting, Ju Ziyi, Li Zhengming, et al. GraphPlanner: Floorplanning with graph neural network[J]. ACM Transactions on Design Automation of Electronic Systems, 2023, 28(2): 1−24

[49] Zhong Ruizhe, Du Xingbo, Kai Shixiong, et al. FlexPlanner: Flexible 3D floorplanning via deep reinforcement learning in hybrid action space with multi-modality representation[C]//Proc of the 38th International Conf on Neural Information Processing Systems: Vol. 37. Red Hook: Curran Associates Inc., 2024: 49252−49278

[50] Amin F, Rouf N, Pan T, et al. Large reasoning models for 3D floorplanning in EDA: Learning from imperfections[J]. arXiv preprint, arXiv. 2406.10538, 2024

[51] Mohammed H U H, Ying D. Reinforcement learning for automated chip floorplanning and routing optimization[J]. Webology, 2024, 21(2): 1−28

[52] Xu Qi, Geng Hao, Chen Song, et al. GoodFloorplan: Graph convolutional network and reinforcement learning-based floorplanning[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, 41(10): 3492−3502

[53] Guan Wenbo, Tang Xiaoyan, Lu Hongliang, et al. LSTM-characterized approach for chip floorplanning: Leveraging HyperGCN and DRQN[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025, 44(2): 709−722

[54] Wang Shikai, Zhang Haodong, Zhou Shiji, et al. Chip Floorplanning optimization using deep reinforcement learning[J]. Spectrum of Research, 2024, 4(2): 1−15

[55] Ajayi T, Chhabria V, Fogaça M, et al. Toward an open-source digital flow: First learnings from the OpenROAD project[C]//Proc of the 56th Annual Design Automation Conf. New York: ACM, 2019: 1−7

[56] Cheng Chung-Kuan, Kahng A, Kundu S, et al. Assessment of reinforcement learning for macro placement[C]//Proc of the 2023 Int Symp on Physical Design. New York: ACM, 2023: 158–166

[57] Wang Bowen, Shen Guibao, Li Dong, et al. LHNN: Lattice hypergraph neural network for VLSI congestion prediction[C]//Proc of the 59th ACM/IEEE Design Automation Conf. New York: ACM, 2022: 1297−1302

[58] Liu Siting, Sun Qi, Liao Peiyu, et al. Global placement with deep learning-enabled explicit routability optimization[C]//Proc of the Design, Automation & Test in Europe Conf & Exhibition (DATE). Piscataway, NJ: IEEE, 2021: 1821−1824

[59] Lin Yibo, Dhar Shounak, Li Wuxi, et al. DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement[C]//Proc of the 56th Annual Design Automation Conf. New York: ACM, 2019: 1−7

[60] Alhyari A, Shamli A, Abuwaimer Z, et al. A deep learning framework to predict routability for FPGA circuit placement[C]// Proc of the 29th Int Conf on Field Programmable Logic and Applications (FPL). Piscataway, NJ: IEEE, 2019: 334−341

[61] Li Wenchao, Wu Hongxi, Liu Duanxiang, et al. Differentiable net-moving and local congestion mitigation for routability-driven global placement[C]//Proc of the 62nd ACM/IEEE Design Automation Conf. New York: ACM, 2025: 1−6

[62] Cheng Ruoyu, Yan Junchi. On joint learning for solving placement and routing in chip design[C]//Advances in Neural Information Processing Systems: Vol. 34. Red Hook: Curran Associates, Inc., 2021: 16508−16519

[63] Chiang Cheng Yu, Chiang Yi Hsien, Lan Chao Chi, et al. Mixed-size placement prototyping based on reinforcement learning with semi-concurrent optimization[C]//Proc of the 30th Asia and South Pacific Design Automation Conf. New York: ACM, 2025: 893−899

[64] Chen Lili, Lu K, Rajeswaran A, et al. Decision Transformer: Reinforcement learning via sequence modeling[J]. Advances in neural Information Processing Systems, 2021, 34: 15084−15097

[65] Hou Yunbo, Ye Haoran, Yang Shuwen, et al. TransPlace: Transferable circuit global placement via graph neural network[J]. arXiv preprint, arXiv. 2501.05667, 2025

[66] Chen Yan Jen, Hsieh Cheng Hsiu, Su Po Han, et al. Mixed-size 3D analytical placement with heterogeneous technology nodes[C]//Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM, 2024: 1−6

[67] Hsiao Hao Hsiang, Lu Yi Chen, Vanna-Iampikul P, et al. DCO-3D: Differentiable congestion optimization in 3D ICs[C]//Proc of the 62nd ACM/IEEE Design Automation Conf. New York: ACM, 2025: 1−7

[68] Zhao Yuxuan, Liao Peiyu, Yu Bei. 3D-Flow: Flow-based standard cell legalization for 3D ICs[C]//Proc of the 62nd ACM/IEEE Design Automation Conf. New York: ACM, 2025: 1−7

[69] Guo Zizheng, Lin Yibo. Differentiable-timing-driven global placement[C]//Proc of the 59th ACM/IEEE Design Automation Conf. New York: ACM, 2022: 1315−1320

[70] He Xu, Wang Yushan, Yang Chenjing, et al. Efficient timing prediction and optimization using derivable gradient boosting machine model at placement stage[J]. ACM Transactions on Design Automation of Electronic Systems, 2026, 31(6): 1−19

[71] Du Yufan, Guo Zizheng, Lin Yibo, et al. Fusion of global placement and gate sizing with differentiable optimization[C]//Proc of the 43rd IEEE/ACM Int Conf on Computer-Aided Design. New York: ACM, 2024: 1−7

[72] Ye Yuyang, He Mingwei, Ren Lizheng, et al. Truly pre-routing timing prediction via considering power delivery network[C]//Proc of the 62nd ACM/IEEE Design Automation Conf. New York: ACM,

2025: 1−6

[73] Esmaeili P, Martin T, Areibi S, et al. Guiding FPGA detailed placement via reinforcement learning[C]//Proc of the IFIP/IEEE 30th Int Conf on Very Large Scale Integration. Piscataway, NJ: IEEE, 2022: 1−8

[74] Lim D, Park H. Graph neural network-based detailed placement optimization framework[C]//Proc of the 25th Int Symp on Quality Electronic Design. Piscataway, NJ: IEEE, 2024: 1−8

[75] Jiang Xun, Chai Zhuomin, Zhao Yuxiang, et al. CircuitNet 2.0: An advanced dataset for promoting machine learning innovations in realistic chip design environment[C]//Proc of the 12th Int Conf on Learning Representations. Vienna: ICLR, 2023: 1−26

[76] Kahng A, Kang S, Kundu S, et al. PPA-relevant clustering-driven placement for large-scale VLSI designs[C]//Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM, 2024: 1−6

[77] Mangalagiri P, Qian L, Zafar F, et al. CDLS: Constraint driven generative AI framework for analog layout synthesis[C]//Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM, 2024: 1−7

[78] Wang Zhihai, Geng Zijie, Tu Zhaojie, et al. Benchmarking end-to-end performance of AI-based chip placement algorithms[J]. arXiv preprint, arXiv: 2407.15026, 2024

[79] Chhabria V, Gopalakrishnan V, Kahng A, et al. Strengthening the foundations for IC physical design and ML EDA research[C]//Proc of the 43rd IEEE/ACM Int Conf on Computer-Aided Design. New York: ACM, 2024: 1−7

[80] Liu Lixin, Fu Bangqi, Wong Martin D F, et al. Xplace: An extremely fast and extensible global placement framework[C]//Proc of the 59th ACM/IEEE Design Automation Conf. New York: ACM, 2022: 1309−1314

[81] Cao Peng, Li Zhi, Ding Wenjie. A GNN-based placement optimization guidance framework by physical and timing prediction[J]. Electronics, 2025, 14(2): 329

[82] Ding Wenjie, Zhang Zhanhua, He Guoqing, et al. A physical and timing aware placement optimization framework based on graph neural network[C]//Proc of the 43rd IEEE/ACM Int Conf on Computer-Aided Design. New York: ACM, 2024: 1−7

[83] Yoon J, Lee J, Kim D, et al. ParaFormer: A hybrid graph neural network and Transformer approach for pre-routing parasitic RC prediction[C]//Proc of the 30th Asia and South Pacific Design Automation Conf. New York: ACM, 2025: 513−519

[84] Sechen C, Sangiovanni-Vincentelli A. The TimberWolf placement and routing package[J]. IEEE Journal of Solid-State Circuits, 1985, 20(2): 510−522

[85] Eisenmann H, Johannes F. Generic global placement and floorplanning[C]//Proc of the 35th Annual Design Automation Conf. New York: ACM, 1998: 269−274

[86] Chen Tung Chieh, Jiang Zhe Wei, Hsu Tien Chang, et al. NTUplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27(7): 1228−1240

[87] Lu Jingwei, Chen Pengwen, Chang Chin Chih, et al. ePlace: Electrostatics based placement using Nesterov's method[C]//Proc of the 51st Annual Design Automation Conf. New York: ACM, 2014: 1−6

[88] Cheng C, Kahng A, Kang I, et al. RePlAce: Advancing solution quality and routability validation in global placement[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2019, 38(9): 1717−1730

[89] Szentimrey H, Al-Hyari A, Foxcroft J, et al. Machine learning for congestion management and routability prediction within FPGA placement[J]. ACM Transactions on Design Automation of Electronic Systems, 2020, 25(5): 1−25

[90] Nagaria S, Deb S. Designing of an optimization technique for the prediction of CTS outcomes using neural network[C]//Proc of the IEEE Int Symp on Smart Electronic Systems. Piscataway, NJ: IEEE, 2020: 312−315

[91] Jain A, Das P, Acharyya A. Artificial neural network based post-CTS QoR report prediction[C]//Proc of the IEEE Int Symp on Circuits and Systems. Piscataway, NJ: IEEE, 2022: 682−686

[92] Jain A, Das P, Acharyya A, et al. ANN-based accurate and fast post-route QoR data prediction methodology from pre-clock tree synthesis by skipping CTS and routing[C]//Proc of the IEEE Int Symp on Circuits and Systems. Piscataway, NJ: IEEE, 2024: 1−7

[93] Du Yufan, Guo Zizheng, Jiang Xun, et al. PowPrediCT: Cross-stage power prediction with circuit-transformation-aware learning[C]//Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM, 2024: 1−7

[94] Kahng A, Lin B, Nath S. High-dimensional metamodeling for prediction of clock tree synthesis outcomes[C]//Proc of the ACM/IEEE Int Workshop on System Level Interconnect Prediction. Piscataway, NJ: IEEE, 2013: 1−6

[95] Lu Yi Chen, Lee J, Agnesina A, et al. A clock tree prediction and optimization framework using generative adversarial learning[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, 41(9): 3104−3117

[96] Du Yufan, Guo Zizheng, Wang Runsheng, et al. Differentiable physical optimization[C]//Proc of the 44th IEEE/ACM Int Conf on Computer-Aided Design. Piscataway, NJ: IEEE, 2025: 1−7

[97] Wu Jiang, Ni Chunhe, Wang Hongbo, et al. Graph neural networks for efficient clock tree synthesis optimization in complex SoC designs[J]. Applied and Computational Engineering, 2025, 150(1): 101−111

[98] Ray P, Prashant V, Rao B. Machine learning based parameter tuning for performance and power optimization of multisource clock tree synthesis[C]//Proc of the IEEE 35th Int System-on-Chip Conf. Piscataway, NJ: IEEE, 2022: 1−9

[99] Choi B, Kwon Y, Afzaal U, et al. Multisource clock tree synthesis through sink clustering and fast clock latency prediction[C]//Proc of the IEEE Int Symp on Circuits and Systems. Piscataway, NJ: IEEE, 2023: 1−7

[100] Liu Meng, Zhang Zhiye, Wen Jiabao, et al. An approximate symmetry clock tree design with routing topology prediction[C]//

Proc of the IEEE Int Midwest Symp on Circuits and Systems. Piscataway, NJ: IEEE, 2021: 92−96

[101] Bethur N, Vanna P, Zografos O, et al. GNN-assisted back-side clock routing methodology for advance technologies[C]//Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM, 2024: 1−7

[102] Jiang Xun, Lu Haoran, Zhao Yuxuan, et al. A systematic approach for multi-objective double-side clock tree synthesis[J]. arXiv preprint, arXiv: 2503.12512, 2025

[103] Beheshti S, Nazari N, Gubbi K, et al. Advanced reinforcement learning solution for clock skew engineering: Modified Q-table update technique for peak current and IR drop minimization[J]. IEEE Access, 2023, 11(1): 87869−87886

[104] Beheshti S, Vakil A, Manoj S, et al. A reinforced learning solution for clock skew engineering to reduce peak current and IR drop[C]// Proc of the 2021 Great Lakes Symp on VLSI. New York: ACM, 2021: 181−187

[105] Jackson M, Srinivasan A, Kuh E. Clock routing for high-performance ICs[C]//Proc of the 27th ACM/IEEE Design Automation Conf. New York: ACM, 1990: 573−579

[106] Cong J, Kahng A, Robins G. Matching-based methods for high-performance clock routing[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1993, 12(8): 1157−1169

[107] Boese K, Kahng A. Zero-skew clock routing trees with minimum wirelength[C]//Proc of the 5th Annual IEEE Int ASIC Conf and Exhibit. Piscataway, NJ: IEEE, 1992: 17−21

[108] Li Weiguo, Huang Zhipeng, Yu Bei, et al. Toward controllable hierarchical clock tree synthesis with skew-latency-load tree[C]// Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM, 2024: 1−7

[109] Chen Huang Yu, Chang Yao Wen. Global and Detailed Routing[M]//Electronic Design Automation. San Francisco, CA: Morgan Kaufmann, 2009: 687−749

[110] Chen Lin, Li Yuxuan, Ding Hu. Achieving simultaneous buffering and steiner tree synthesis via harmonic based reinforcement learning[C]//Proc of IEEE/ACM Int Conf On Computer Aided Design. Piscataway, NJ: IEEE, 2025: 1−7

[111] Utyamishev D, Partin-V K. Late breaking results: A neural network that routes ICs[C]//Proc of the 57th ACM/IEEE Design Automation Conf. New York: ACM, 2020: 1−6

[112] Liu Jinwei, Chen Gengjie, Young Evangeline F Y. REST: Constructing rectilinear Steiner minimum tree via reinforcement learning[C]//Proc of the 58th ACM/IEEE Design Automation Conf (DAC). Piscataway, NJ: IEEE, 2021: 1135−1140

[113] Chen Liang Ting, Kuo Hung Ru, Li Yih Lang, et al. Arbitrary-size multi-layer OARSMT RL router trained with combinatorial monte-carlo tree search[C]//Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM, 2024: 1200−1206

[114] Chen Zhiyang, Yao Hailong, Yin Xia. PatLabor: Pareto optimization of timing-driven routing trees[C]//Proc of the 62nd ACM/IEEE Design Automation Conf (DAC). New York: ACM, 2025: 1−7

[115] Wang Xianglu, Ding Hu. Towards multi-objective routing: A novel coreset-based transfer learning framework[C]//Proc of the IEEE/ACM Int Conf On Computer Aided Design. Piscataway, NJ: IEEE, 2025: 1−7

[116] Fang Yunkai, Lin Ye Chih, Wang Ting Chi. A fast and robust global router with capacity reduction techniques[C]//Proc of the 29th Asia and South Pacific Design Automation Conf. Piscataway, NJ: IEEE, 2024: 1109−1115

[117] Liu Jinwei, Pui C, Wang Fangzhou, et al. CUGR: Detailed-routability-driven 3D global routing with probabilistic resource model[C]//Proc of the 57th ACM/IEEE Design Automation Conf. Piscataway, NJ: IEEE, 2020: 778−784

[118] Zeng Zhisheng, Liu Jikang, Huang Zhipeng, et al. Net resource allocation: A desirable initial routing step[C]//Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM, 2024: 1−6

[119] Hu Jiawei, Vanna-Iampikul P, Zhuang Zhen, et al. GNN-MLS: Signal routing in mixed-node 3D ICs through GNN-Assisted metal layer sharing[C]//Proc of the 62nd ACM/IEEE Design Automation Conf. New York: ACM, 2025: 1−6

[120] Liu He, Zeng Zhisheng, Tao Simin, et al. AiTPO: KAN-UNet heterogeneous network for timing prediction and optimization at global routing[J]. ACM Transactions on Design Automation of Electronic Systems, 2025, 1(1): 3735639

[121] Liu Siting, Wang Ziyi, Liu Fangzhou, et al. Sign-off timing considerations via concurrent routing topology optimization[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025, 44(5): 1942−1953

[122] Zhao Chunyuan, Guo Zizheng, Wang Rui, et al. HeLEM-GR: Heterogeneous global routing with linearized exponential multiplier method[C]//Proc of the 43rd IEEE/ACM Int Conf on Computer-Aided Design. New York: ACM, 2024: 1−6

[123] Lin Shiju, Xiao Liang, Liu Jinwei, et al. InstantGR: Scalable GPU parallelization for global routing[C]//Proc of the 43rd IEEE/ACM Int Conf on Computer-Aided Design. New York: ACM, 2024: 1−6

[124] Zhang Ping, Yao Pengju, Li Xingquan, et al. V-GR: 3D global routing with via minimization and multi-strategy rip-up and rerouting[C]//Proc of the 29th Asia and South Pacific Design Automation Conf. Piscataway, NJ: IEEE, 2024: 963−968

[125] Li Wei, Liang Rongjian, Agnesina A, et al. DGR: Differentiable global router[C]//Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM, 2024: 1−7

[126] Li Yifan, Liu Ruizhi, Zeng Zhisheng, et al. AiDRC: Accelerating detailed routing by AI-driven design rule violation prediction and checking[J]. ACM Transactions on Design Automation of Electronic Systems, 2025(1): 3769306

[127] Zhao Chunyuan, Wang Jiarui, Jiang Xun, et al. GTA: GPU-accelerated track assignment with lightweight lookup table for conflict detection[C]//Proc of the IEEE/ACM Int Conf On Computer Aided Design. Piscataway, NJ: IEEE, 2025: 1−6

[128] Jiang Ying Jie, Fang Shao Yun. Concurrent detailed routing with pin pattern re-generation for ultimate pin access optimization[C]//Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM,

2024: 1−7

[129] Zhou Zhonghua, Zhu Ziran, Chen Jianli, et al. Congestion-aware global routing using deep convolutional generative adversarial networks[C]//Proc of ACM/IEEE 1st Workshop on Machine Learning for CAD (MLCAD). Piscataway, NJ: IEEE, 2019: 1−6

[130] Liu Biao, Qiao Congyu, Xu Ning, et al. Variational label-correlation enhancement for congestion prediction[C]//Proc of the 29th Asia and South Pacific Design Automation Conf. Piscataway, NJ: IEEE, 2024: 466−471

[131] Lin Zhenkun, Lin Yibo, Liu Genggeng, et al. URoute: Universal routability prediction[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025, 1(1): 1−6

[132] Zhang Xinyun, Zhu Binwu, Liu Fangzhou, et al. Disentangle, align and generalize: Learning a timing predictor from different technology nodes[C]//Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM, 2024: 1−6

[133] Xu Peng, Chen Guojin, Zhu Keren, et al. Performance-driven analog routing via heterogeneous 3DGNN and potential relaxation[C]//Proc of the 61st ACM/IEEE Design Automation Conf. New York: ACM, 2024: 1−7

[134] Hsu Chen Hao, Pan David Z, Perron L, et al. TransRoute: A novel hierarchical transistor-level routing framework beyond standard-cell methodology[C]//Proc of the 62nd ACM/IEEE Design Automation Conf. New York: ACM, 2025: 1−7

[135] Wu Hongxi, Li Xingquan, Chen Liang, et al. Delay-driven rectilinear steiner tree construction[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025, 44(5): 1928−1941

[136] Qi Zhongdong, Hu Shizhe, Peng Qi, et al. A high performance detailed router based on integer programming with adaptive route guides[C]//Proc of the 29th Asia and South Pacific Design Automation Conf (ASP-DAC). Piscataway, NJ: IEEE, 2024: 975−980

[137] Bihler T, Blankenburg D. Dynamic local usage: An accurate model for usage of tile-internal wiring in global routing[C]//Proc of the 62nd ACM/IEEE Design Automation Conf. New York: ACM, 2025: 1−7



Liu Ruizhi, born in 1996. PhD. His main research interests include open-source EDA, AI for EDA, algorithm design, and combinatorial optimization.

刘睿之, 1996 年生。博士。主要研究方向为开源 EDA、AI 启发的 EDA、算法设计、组合优化。



Li Jiepeng, born in 2000. Master candidate. His main research interests include algorithm design, large language model, and reinforcement learning.

李捷鹏, 2000 年生。硕士研究生。主要研究方向为算法设计、大语言模型、强化学习。



Wang Rui, born in 2001. PhD candidate. Her main research interests include algorithm design, machine learning, deep learning, and combinatorial optimization.

王　睿, 2001 年生。博士研究生。主要研究方向为算法设计、机器学习、深度学习、组合优化。



Sui Jingyan, born in 1982. PhD. Her main research interests include algorithm design, machine learning, deep learning, and combinatorial optimization and bioinformatics.

隋京言, 1982 年生。博士。主要研究方向为算法设计、机器学习、深度学习、组合优化和生物信息。



Li Xingquan, born in 1990. PhD, associate professor. His main research interests include electronic design automation (EDA), AI for EDA, and AI for algorithm design.

李兴权, 1990 年生。博士，副研究员。主要研究方向为 EDA、AI 启发的 EDA、AI 启发的算法设计。



Xie Biwei, born in 1987. PhD, associate professor. His main research interests include open-source EDA, open-source chip, high performance computing, and computer architecture.

解壁伟, 1987 年生。博士，副研究员。主要研究方向为开源 EDA、开源芯片、高性能计算、计算机体系结构。



Yu Chungong, born in 1979. Master. His main research interests include bioinfomatics and protein structure prediction.

于春功, 1979 年生。硕士。主要研究方向为生物信息学、蛋白质结构预测。



Bu Dongbo, born in 1973. PhD, professor, PhD supervisor. His main research interests include algorithm design, bioinformatics, protein structure prediction, and deep learning.

卜东波, 1973 年生。博士，研究员，博士生导师。主要研究方向为算法设计、生物信息学、蛋白质结构预测、深度学习。