**人工智能（AI）在详细布线中的应用**

# 1.背景

详细布线在集成电路设计中至关重要，直接影响芯片的性能、可靠性和制造成本。合理的布线能够优化性能，减少信号传输延迟，降低噪声干扰，从而提升电路频率和速度。在确保可靠性方面，详细布线需要满足电气和热设计规则，避免过热和保证足够的电流承载能力，以确保电路稳定性和寿命。通过优化布局，详细布线还能够控制制造成本，减少制造过程中的复杂性和潜在缺陷，降低层间过孔数量。详细布线必须严格遵循设计规则，确保所有连接符合要求，避免制造问题。它与全局布线协同工作，在提供大致连接框架的基础上进行精确路径规划，确保高效布局。随着技术节点的缩小，芯片特征尺寸变小、集成度增高，详细布线面临更大挑战，传统方法难以满足要求，需要新的AI技术解决方案。创新设计也常需非传统布线策略，AI技术可以帮助实现独特设计思路，推动技术进步。

人工智能（AI）技术，特别是机器学习和深度学习，在提高集成电路（IC）设计中的详细布线效率和质量方面展现出巨大潜力。AI通过自适应学习能力和动态优化，能够适应复杂的设计规则，并根据实时数据调整布线策略。此外，AI擅长处理大规模数据集，从历史设计中学习模式，提取关键特征，进而提升布线决策的质量。创新的搜索策略，如强化学习和并行探索，使AI能够发现传统算法可能忽略的布线路径，快速生成多样化的解决方案。精确的预测模型能够预见设计规则违规和拥堵区域，提前进行预防性调整。AI的自动化和智能化特性减少了人工干预，提高了设计效率，同时提供智能决策支持。AI系统通过在线学习和反馈循环不断优化布线策略，展现出强大的鲁棒性和泛化能力，能够处理不同类型的设计问题。与现有EDA工具的无缝集成，使AI能够协同工作，提升设计流程的整体性能。AI技术不仅能提升现有详细布线方法的性能，还能开辟新的研究方向，推动IC设计领域的创新和发展。随着AI技术的不断进步，它将在详细布线领域发挥越来越关键的作用。

在本章节，我们将重点讨论详细布线（Detailed Routing, DR）领域中人工智能（AI）技术的应用，主要集中在以下四个方面：引脚访问（pin access）、网络顺序（net order）、轨道分配（track assignment）以及设计规则检查（DRC）预测。

# 深度学习预测算法在详细布线中的应用

这一章写引脚可访问性预测和DRC热点预测

（1）引脚访问

# **2.1传统引脚访问**

引脚访问（Pin Access）是集成电路设计中的一个重要概念，它涉及到芯片上每个逻辑单元（例如标准单元或宏单元）的引脚如何被连接到电路的其他部分。在复杂的芯片设计中，确保每个引脚都能够被有效地访问并连接是至关重要的，因为这直接关系到布局布线的效率和最终芯片的性能。

引脚访问分析包括确定引脚的位置、评估引脚的可达性以及生成引脚访问模式。这些模式必须遵守设计规则，同时考虑到制造过程中的技术限制。在现代的集成电路设计流程中，引脚访问分析通常需要自动化工具来执行，这些工具可以生成符合设计规则的引脚访问点，并通过动态规划等算法优化引脚的访问模式，以减少布局布线过程中的冲突和违规，提高设计的整体质量。下面图1是一个引脚访问问题的例子。

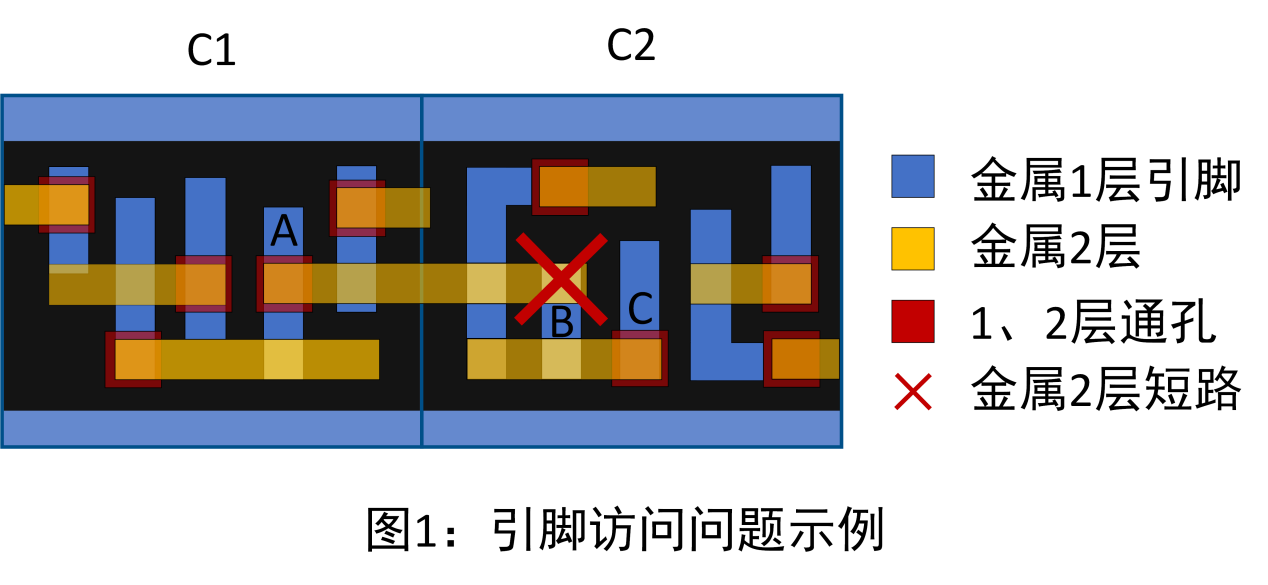


图1展示了标准单元在更低金属层级上由于引脚密度高、引脚可访问性低和路由资源有限而导致的路由问题。以单元C2为例，其引脚的可访问性受到了邻近单元C1的影响。从图中可以观察到，由于来自引脚A和引脚C的金属2层（M2）路由段，引脚B的访问点被阻塞，如果在引脚B上放置一个金属2层的过孔（via12），将会引起M2短路的设计规则违规（DRV）。这个例子说明了引脚可访问性不仅由单元布局设计决定，还受到邻近单元的强烈影响。随着半导体工艺节点的不断缩小，标准单元变得更小，单元数量急剧增加，这种影响变得更加显著，导致在低层金属层上的路由问题更加严重。

# **2.2 AI在引脚访问中的应用**

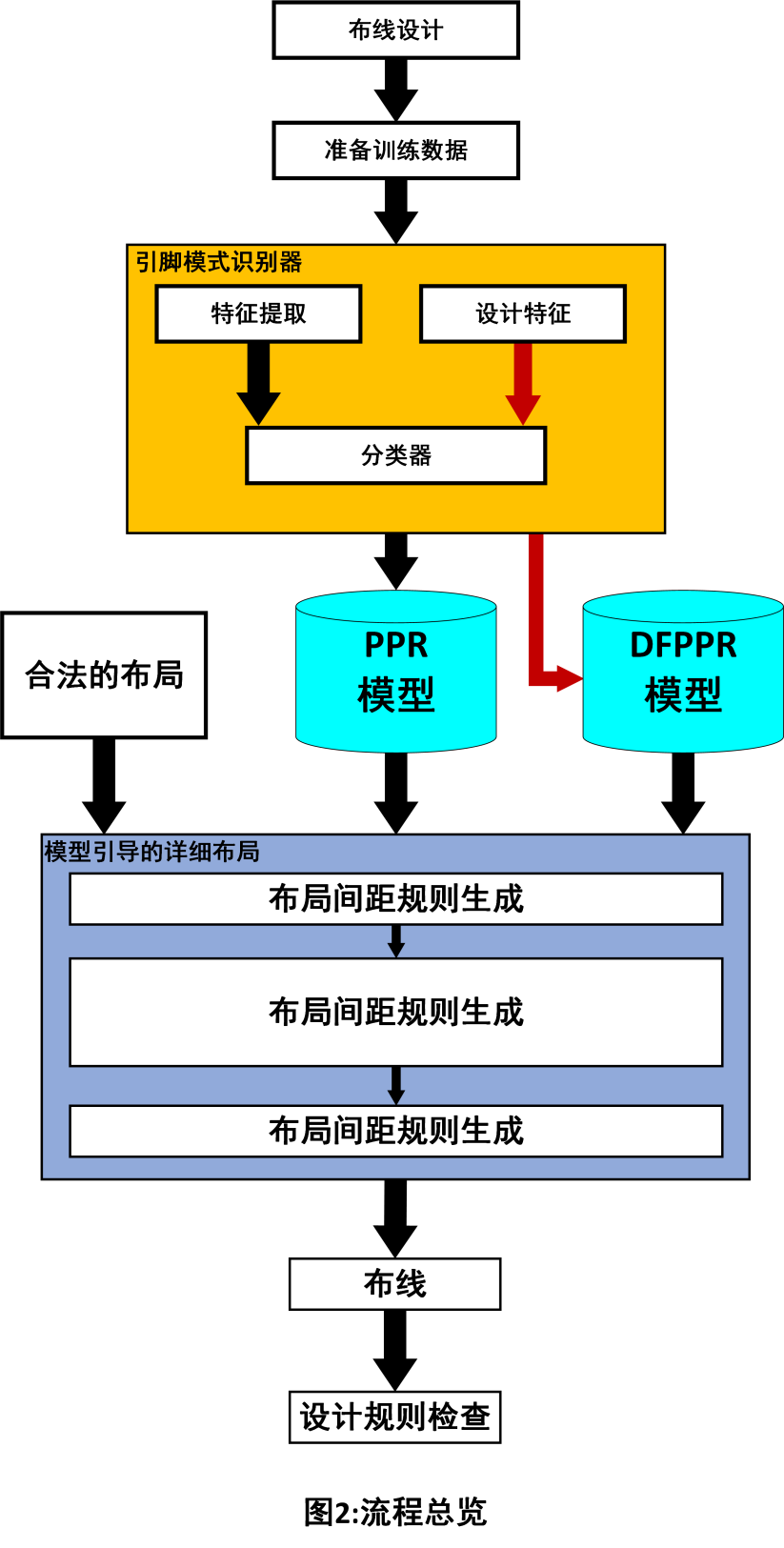
**基于深度学习的引脚模式识别的引脚可及性预测与优化**

### **（1）预测引脚可访问性**

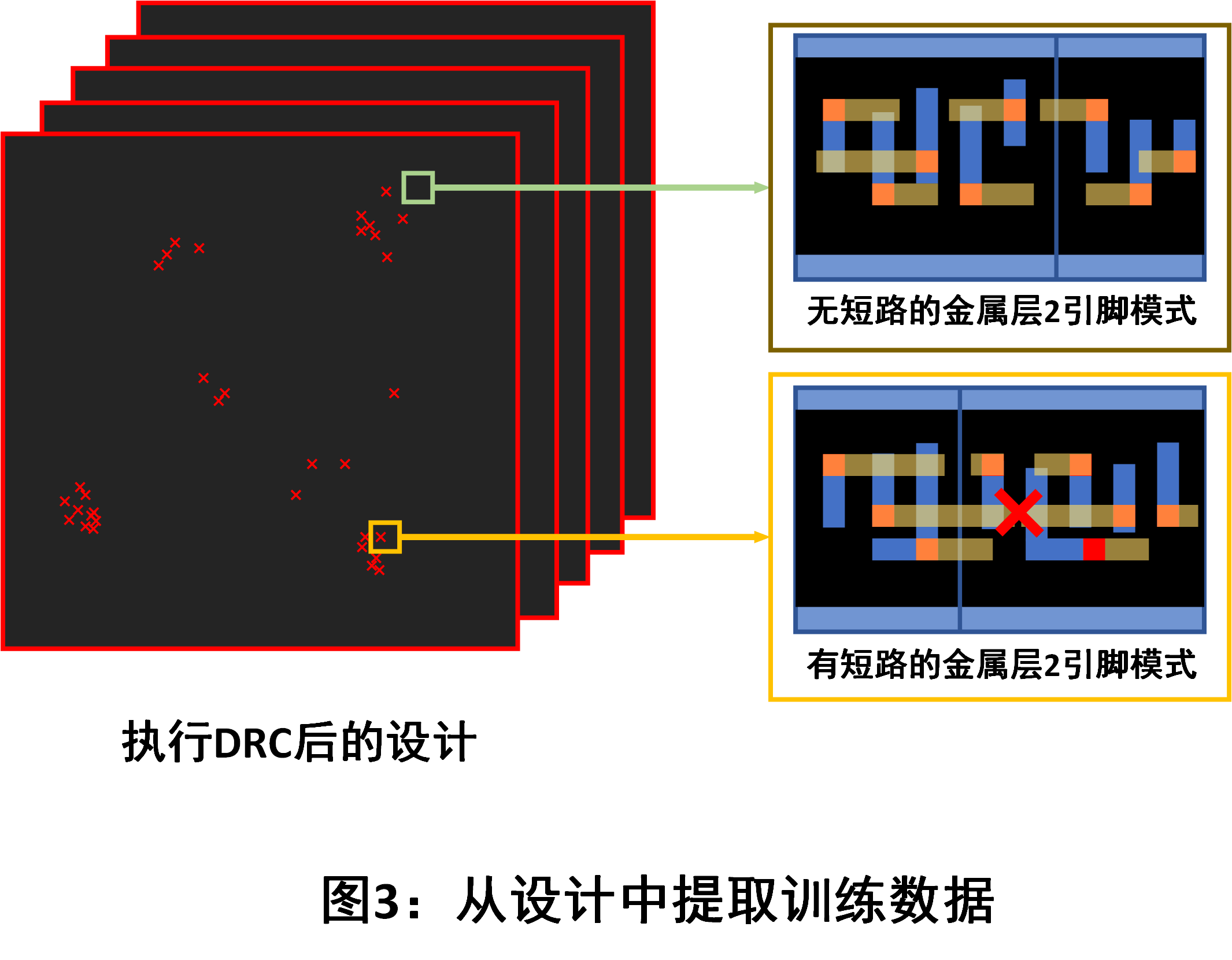
现有工作中使用人工智能技术来预测哪些引脚易于路由访问，哪些存在问题。这通过分析引脚之间的空间关系和布局特征来实现。卷积神经网络（CNN）、支持向量机（SVM）、图神经网络（GNN）等模型能够自动识别引脚布局中的关键特征，从而预测哪些引脚会在布线过程中遇到困难，并提供相应的优化建议。这些方法大大提高了预测的准确性和效率，使设计者能够在早期阶段识别和解决潜在问题。

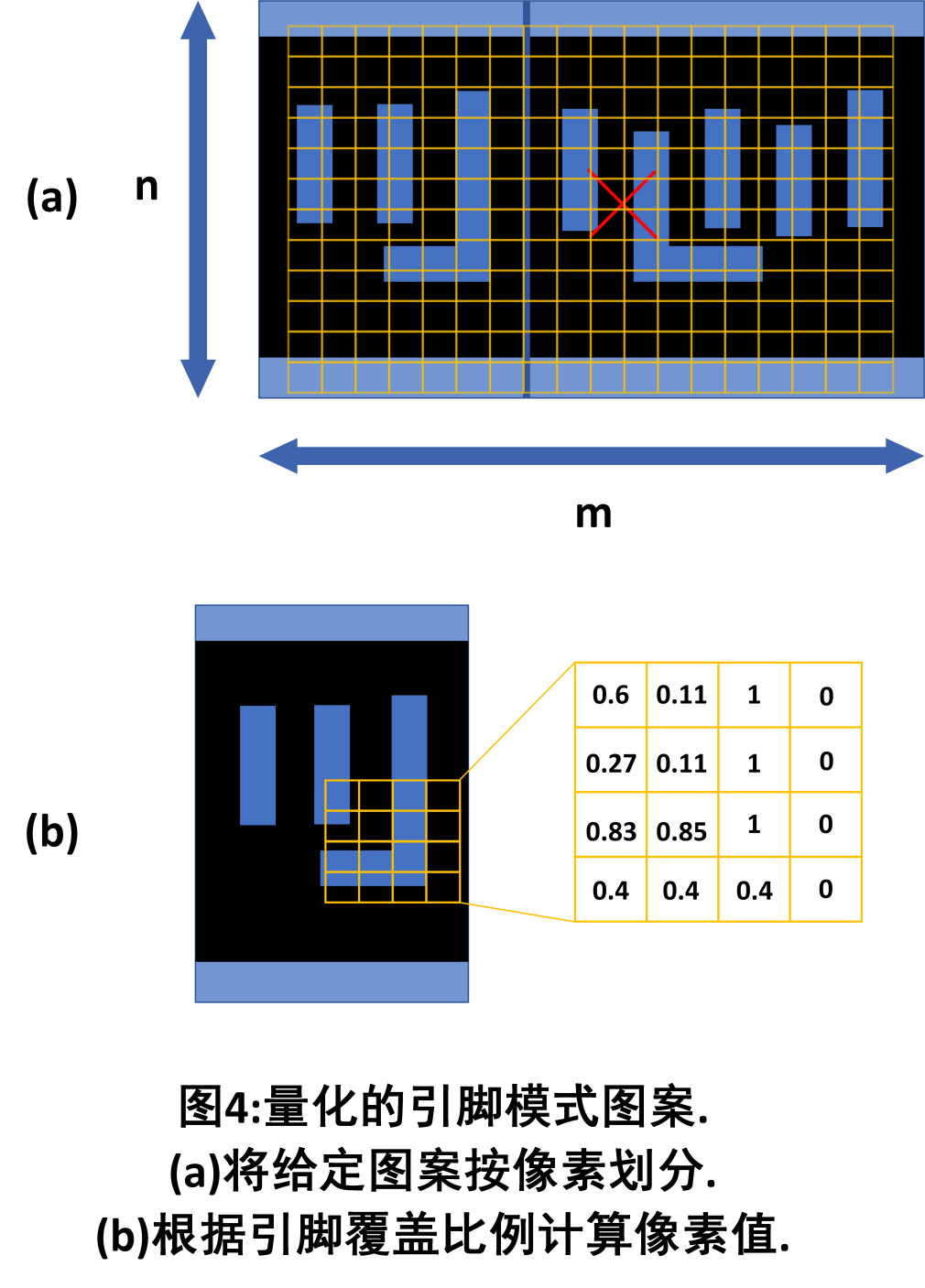
**训练数据准备**

所提出方法的概述如图2所示，可以划分为三个阶段：训练数据准备、模型训练和模型引导的详细放置。提出了两种模型：引脚图案识别（PPR）模型使用引脚图案作为唯一特征来预测由不良引脚可访问性引起的设计规则违规（DRVs），以及设计特征感知的引脚图案识别（DFPPR）模型，该模型采用更多设计特征来进一步提高预测准确性。然后提出了模型引导的详细放置，以避免在详细放置期间生成具有不良引脚可访问性的引脚图案。所提出的模型引导的详细放置由三个阶段组成：首先通过应用PPR模型生成放置间距规则，这些规则作为详细放置的硬约束。然后，应用基于动态规划的详细放置算法来最小化每个单元行插入的放置阻塞数量。最后，使用单元位移细化进一步优化放置。



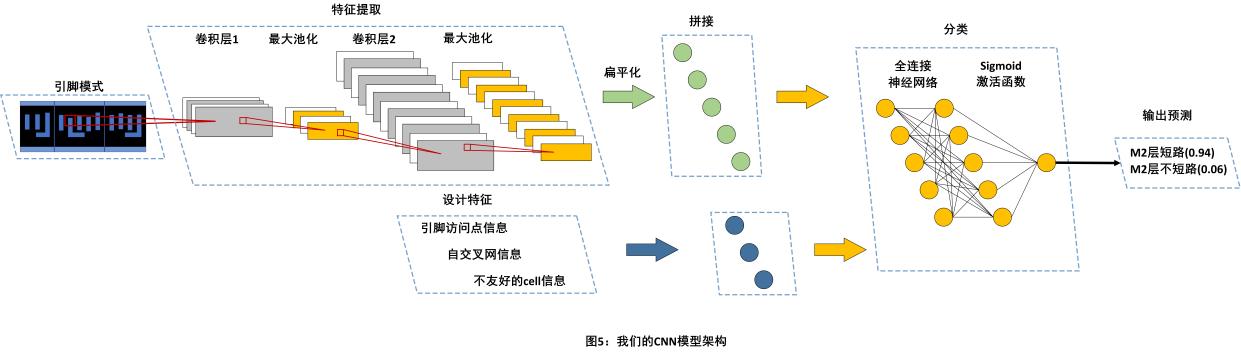
为了应用监督学习，有必要获取训练数据的所有标签。由于我们的目标是识别给定的引脚图案是否会诱发M2短路，我们需要生成一些布线设计以收集足够数量的引脚图案作为训练数据。如图5所示，我们首先生成一些报告M2短路位置的布线设计。随后，我们提取所有有M2短路的引脚图案和一些没有M2短路的引脚图案来构建我们的训练数据集。将M2短路定位在引脚图案的中心，可以通过分别设置图像的宽度和高度为2×单行高度和单行高度来获得引脚图案图像，如图7(a)所示。通过使用上述宽度设置，一个引脚图案图像通常可以覆盖至少两个单元，因此可以考虑相邻单元影响的引脚可访问性。





**模型训练**

在本节中，为了DRV预测，提出了两种深度学习模型。第一种是引脚图案识别（PPR）模型，其训练流程如图6所示，不包含设计特征模块。在特征提取阶段，每个输入的引脚图案图像被送入卷积神经网络（CNN）以提取代表性特征。然后，所有特征被展平并连接到全连接神经网络进行分类。另一种是设计特征感知的引脚图案识别（DFPPR）模型。与PPR模型不同，在被送入分类模块之前，由CNN计算得到的代表性特征将与其他设计相关特征结合。下面详细描述所提出的两种模型。



**引脚图案识别（PPR）模型**

所提出的PPR模型架构由特征提取块和分类块组成。在特征提取之前，需要对输入的引脚图案图像进行量化。首先将每个引脚图案分割为m×n像素，如图7(a)所示。每个像素的宽度和高度都设置为M1层的最小间距，以防止一个像素被两个不同的引脚占据。每个像素的值通过相应的被引脚覆盖的比率来计算。如图7(b)所示，每个像素的被引脚覆盖比率是通过引脚覆盖区域与像素面积的比值来计算的。之后，这些量化的像素值被送入特征提取阶段。特征提取阶段包括两个卷积层和两个最大池化层。卷积层提取给定引脚图案的关键特征，这些特征由可训练的滤波器加权。最大池化层用于减少参数量，并通过滤除噪声来保留那些代表性的参数。一旦特征提取完成，所有提取的特征被展平并送入分类阶段。分类块由一个全连接神经网络和Sigmoid函数组成。全连接神经网络是一个深度神经网络（DNN），包含多个神经元层，为复杂特征提供更多抽象层次。Sigmoid函数通常应用于二元分类问题的最后一层，用于将输出值缩放到0和1之间。这种缩放可以帮助模型根据预定义的阈值来判断给定的输入引脚图案是否会诱发M2短路。如果输出值不低于0.5，模型将将给定的输入引脚图案视为M2短路候选。

**设计特征感知引脚图案识别（DFPPR）模型**

所提出的DFPPR模型架构大致与PPR模型相同。主要区别在于，可以额外考虑从给定设计中提取的一些特征，这显示了用户可以采用与DRV发生相关的任何特征与引脚图案的特征结合，以进一步提高预测准确性的极大灵活性。如图6所示，在分类块之前，不仅从引脚图案中提取的特征，还有三个设计特征被连接起来。我们在这项工作中采用的设计特征如下：

- 引脚访问点信息：此特征通过输入引脚图案中引脚数量与访问点数量的比率来计算。

- 自交叉网络信息：自交叉网络是由两个局部网络的飞线相交引起的。飞线是连接两个引脚左下角访问点的直线。此特征通过输入引脚图案中自交叉网络数量与局部网络数量的比率来计算。

- 不友好单元信息：此特征受到[1]中使用的“不友好单元”特征的启发。在布线设计中，我们首先计算每个库单元诱发M2短路的频率。然后，此特征是通过引脚图案覆盖的单元频率之和与所有单元总频率的比率来获得的。

**模型指导的详细布局**

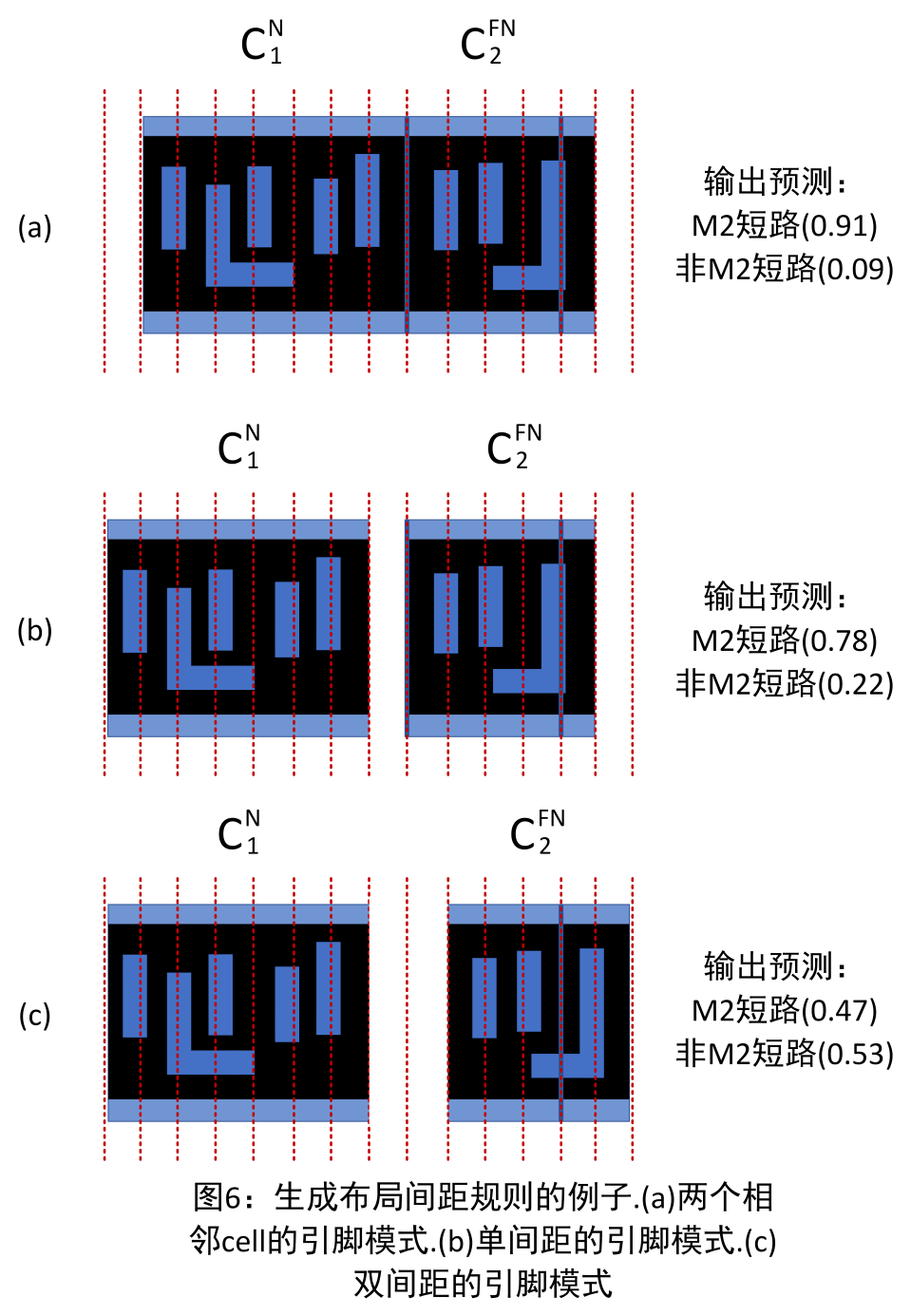
基于提出的PPR模型，我们提出了一种模型指导的详细布局算法，以指导详细布局器避免生成高概率导致M2短路的引脚模式。提出的模型指导的详细布局包含三个阶段：（1）布局间距规则生成，（2）基于动态规划的布局阻塞插入，（3）单元位置调整，详细内容如下。

布局间距规则生成

尽管提出的PPR模型可以精确识别引脚模式是否会导致M2短路，用户不能简单地直接使用它在物理设计过程中避免产生这些不良的引脚模式。因此，开发一个基于预训练模型的DRV感知详细布局器是很有必要的。我们的想法是使用提出的PPR模型生成一组所需的单元间距规则，这是一项一次性的过程，并且规则集可以应用于使用相同单元库的任何设计中。图8展示了布局间距规则生成的示例。假设我们有一个引脚模式，由方向为N的（）和方向为FN的（）组成，预训练模型预测会出现M2短路，如图8(a)所示。在不移动单元和的情况下，输出预测值为0.94，这可以视为发生M2短路的概率。为了防止这种不良引脚模式在给定设计中出现，我们尝试迭代地增加单元之间的间距，直到模型预测它们具有足够的引脚可达性。如图8(b)所示，我们首先将单元向左移动一个位置，并将修改后的引脚模式输入到预训练模型中。输出预测值为0.78，仍然大于默认阈值0.5。因此，我们继续将单元向右移动一个位置，如图8(c)所示。更新后的引脚模式的输出预测值现在为0.47，小于默认阈值。因此，我们将单元和之间的所需间距规则设为两个位置。基于上述思路，一组间距规则将生成每对单元组合之间的规则。以下公式给出了单元库中具有方向m的第i个单元（）和方向n的第j个单元（）之间所需的布局位置数（RPS）。



以图8(c)为例，由于和之间的所需间距等于两个布局位置的宽度，我们将这一对单元的规则设为。



**基于动态规划的布局阻塞插入**

在获得每对单元之间的最小布局位置数后，我们将这些间距规则整合到详细布局器中进行优化。本文提出了一种基于动态规划的布局阻塞插入方法，以在考虑单元方向的情况下，最小化单元行中插入的布局阻塞的总量。首先在表1中定义了一些符号。

每对单元之间所需的最小布局阻塞宽度可以通过公式(1)获得。例如，如果我们有一个间距规则，则和之间所需的最小布局阻塞宽度将为1个位置。为了在详细布局过程中最小化设计规则违规（DRV）的数量，我们提出了一个双路径图G = {C, PB, S, T}，其结构如图9所示。源节点S和目标节点T分别创建在图的最左端和最右端。对于每个单元，我们只能使用两个方向集合{N, FN}或{S, FS}之一，以不违反电源/接地轨约束（我们在图9中以N, FN为例）。具有方向N的单元节点被创建并放置在线1（L1）上，具有方向FN的单元节点被创建并放置在线2（L2）上。四个不同的布局阻塞节点从连接到，因为对于一对相邻单元，有四种方向组合，这给出了每对单元组合之间的最小所需间距。

基于上述符号和提出的双路径图，我们有以下公式，通过以下递推关系来计算和的最小成本：



### **减少设计规则违规（DRVs）**

AI技术还用于预测并最小化违反设计规则的情况，如短路或间距问题。通过机器学习模型，设计规则检查（DRC）热点可以在详细布线之前被识别。GNN和U-Net等模型能够预测布局中可能出现的DRC违规点，并提出相应的布局调整建议，从而在布线阶段减少违规发生的几率。这种方法提高了设计合规性，减少了后期手动修复的工作量。

通过求解上述公式，可以分别获得每个标准单元的最佳方向以及每对连续单元之间所需的最小布局阻塞宽度。基于动态规划的算法的计算复杂度为O(RC)，其中R是单元行的数量，C是每行中的最大单元数。

1. **DRC热点区域预测（DRC Hot Area Prediction）**

在现代集成电路设计中，先进工艺节点引入的新复杂设计规则使得芯片的实现变得非常具有挑战性。特别是，复杂的设计规则对物理设计提出了很大的要求，在进行单元放置和线路布线的耗时过程中需要进行大量的迭代，以清除所有的设计规则违规（DRV），确保最终的布线符合设计规则。因此，在放置阶段，如果我们能够高度确信地识别出可能在布线阶段出现的DRC（设计规则检查）热点区域，我们可以更加关注那些DRC热点区域中的单元布局，以便迭代的布局和布线过程能够迅速收敛到无DRV的状态。

此部分通常在详细布局（Detailed Placement）部分调整，在布局阶段，如果我们能够高度置信地识别出可能在布线阶段出现的设计规则检查（DRC）热点，我们可以更加关注这些DRC热点中的单元布局，因此在此进行讲解。

**DRC概念**

DRC（Design Rule Check）是设计规则检查的缩写，它是在集成电路设计中进行的一项重要验证步骤。DRC检查的目的是确保设计满足制造工艺的规则和约束。在设计过程中，集成电路设计人员需要遵守一系列由制造工艺定义的规则，以确保电路能够正确地被制造出来并正常工作。

DRC检查通常在布线设计阶段进行，它涉及到布局中的物理结构、线宽、间距、接触、连线等方面的规则。DRC工具会自动扫描设计，检查是否存在不符合规则的情况。如果发现违反规则的部分，DRC工具会生成相应的错误报告，指出具体的问题和位置，设计人员需要根据报告进行修改和调整，直到通过DRC检查。

DRC检查的目标是确保电路的可制造性和可靠性。通过进行DRC检查，可以发现并纠正潜在的制造缺陷，避免后续制造过程中可能出现的问题。同时，DRC检查也有助于提高电路的性能和可靠性，确保信号的传输和互连满足设计要求。

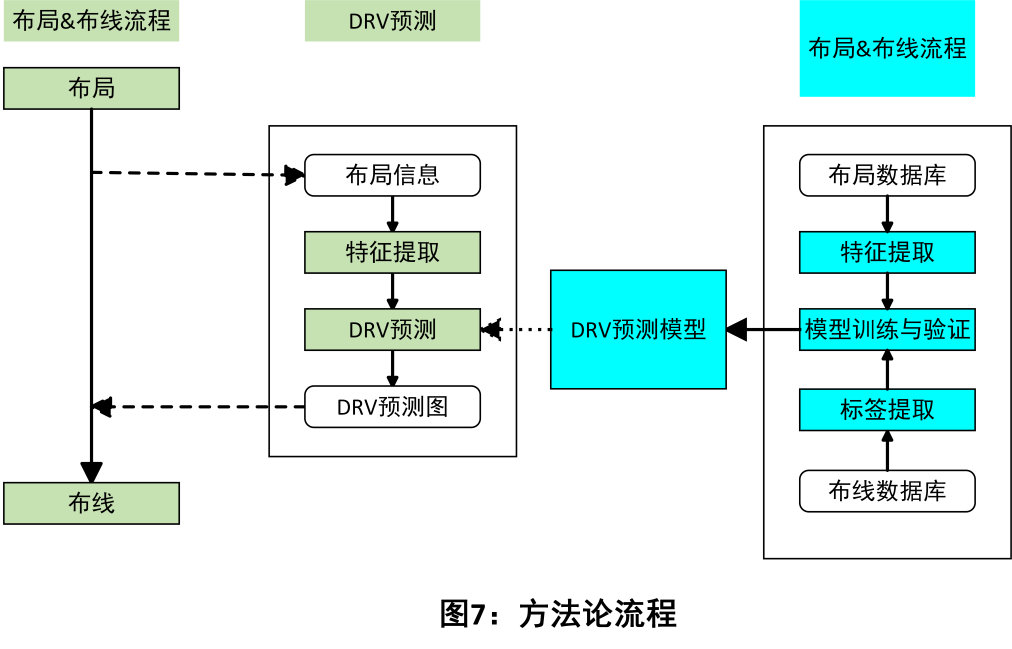
**基于图神经网络和U-Net进行大头针的可访问性和路由拥塞感知的DRC热点预测**

提出了全卷积网络（FCN）[12]来解决语义分割任务，所有层都由卷积层组成。图像识别中的语义分割是指识别输入图像中每个像素的分类。因此，用于语义分割的ML模型应该能够获取任意大小的输入图像，并输出与输入图像相同大小的图像。如果尝试一种基于网格的预测，DRC热点预测问题可以看作是语义分割。U-net [13]是一种基于FCN的网络，在语义分割方面取得了很大的成功。采用如图4所示的结构-编解码器。编码器通过应用一系列的卷积层和池化层，逐渐对输入进行向下采样。解码器还由多个重复的上采样单元组成，它从编码器中获取一个压缩的特征映射作为输入，并将它们恢复到输入的原始大小。每个上采样单元首先通过跳过连接从同一级别上的编码器接收中间特征映射，并将其与先前的上采样单元的输出连接起来。

这个方法创新的基于图神经网络（GNN）和U-Net的机器学习模型PGNN，用于集成电路设计的布局阶段预测设计规则检查（DRC）热点。通过创建引脚邻近图有效模拟单元I/O引脚的空间信息和引脚间的干扰关系，PGNN能够准确捕捉引脚可访问性和布线拥堵对DRC热点的联合影响。在Nangate 15nm库的基准测试中，PGNN在提高预测精度的同时显著减少了推理时间，显示出比现有技术更优异的性能。

我们提出的预测方法论的总体流程如图5所示。首先，整个布局区域被划分为二维（W × H）的网格，其宽度和高度与 G-单元的大小相等。为了准备训练数据，我们在参考电路上进行了布局和布线，详细布线后提取了DRC热点作为基准数据。输出标签也是一个二维的二值图，与输入特征图具有相同的大小，表示哪些网格是DRC热点。

训练好的模型可以在商业的布局和布线流程中采用。在布局完成后，从布局结果中提取输入特征，并通过我们的预测模型生成DRC热点预测结果，这可以在进行布线之前优化布局。



**训练数据准备**

引脚邻近图影响引脚可及性的因素有四个：(1)引脚的形状（或长度）；(2)接引脚网的接近方向；(3)相邻引脚引起的引脚接入干扰；(4)通过网格的网数。为了容纳这些信息，我们在一个局部网格中将引脚信息表述为引脚接近图，其中每个节点表示一个不同的引脚，两个引脚之间的影响由对应引脚的两个节点之间存在的边表示。对于每个网格，假设在中有一组引脚。在引脚邻近图中，每个节点对应一个不同的引脚，将的节点特征表示为，其细节如下：平均x坐标引脚接入点：该特性是引脚中所有接入点的x坐标的平均值。在这种情况下，x坐标是基于网格的相对坐标。例如，如果接入点位于网格的左侧边缘，则x坐标设置为0，如果接入点位于网格的右侧边缘，则x坐标设置为1。这个特征表示的x方向位置。引脚数字矢量：跨金属2轨的引脚上接入点的数量越多，引脚可访问性越好。引脚的引脚数字矢量是一个二进制矢量，它表示单个金属2轨道是否穿过上的接入点。例如，图6(a)中的单元格在网格中有三个引脚。黄色虚线表示金属2轨，网格内共有7条金属2轨，其中在顶部和底部金属2轨上都没有接入点，因此，。

边是从到的有向边，表示在访问时由于的存在而引起的扰动。如果引脚和位于一定的短距离内，这意味着这两个引脚涉及不小的相互作用量，则存在。我们提取了影响两个引脚之间引脚可达性的多种特征，并将这些特征表述为，其细节如下：相对于的访问方向的相对位置：对于和，我们采用[14]提出的方法在四个可能的方向（左、右、上、下）中估计网的接近方向，其中使用FLUTE[15]和边缘移动[16]技术快速生成无线和拥塞驱动的Steiner树。我们称估计方向为主要方向，相反的最后方向，其余的为次要方向。如果位于主方向，则对的可达性影响很大。是一个单热向量，它表示相对于的位置及其接近方向。例如，在图6(c)中，FLUTE估计是从左方向访问的。由于位于的左侧（即主方向），因此)。否则，位于的右侧（即最后一个方向），。和之间的距离：两个引脚位置越近，对彼此的影响越大。特别是，如果两个引脚位置很近，可能会发生行尾或间距规则违反。

重叠接入点磁道比率：该特性是指和都有接入点的金属-2磁道的数量与有接入点的金属-2磁道的数量之比。例如，图6(b)中的引脚在7个金属-2磁道上都有接入点，引脚和在中心的3个金属-2磁道上都有接入点，。

此特征表示对的水平影响量。网格上的水平路由拥塞：网格上的高水平路由拥塞意味着由于通过网格的网的影响，访问网格中的引脚的难度很高。因此，我们采用水平网密度作为引脚接近图边缘的特征。水平净密度的详细计算将在3.3节中描述。利用这些特征，引脚接近图可以有效地表示网格中引脚之间复杂的相互作用。每个网格生成一个独立的引脚接近图，并且对应于两个不同网格的两个图之间没有外部连接。我们在保留位置信息的同时，将所有网格的图转换成一个三维的引脚接近图。

3.3基于网格的特性从网格中提取的特征对预测模型的性能有很大影响。以前的著作[1-7]提出了各种基于网格的特征。通过大量输入特征的密集实验，我们选择了一组显著影响模型性能的代表性特征。这些特性是没有的。引脚密度：该特征表示引脚占用的面积与网格面积的比率。由于大多数引脚都在金属1层上，因此密度分别为金属1引脚和金属2引脚提取。引脚密度值高意味着网格上存在大量引脚，这将导致网格周围密集的路由拥塞。

全局/局域网：全局网是连接网格内引脚的输入或输出边，局域网连接网格内的引脚。我们计算这些网络的数量，并将它们作为输入特征。电网上大量这样的电网意味着发生DRV的可能性很高。

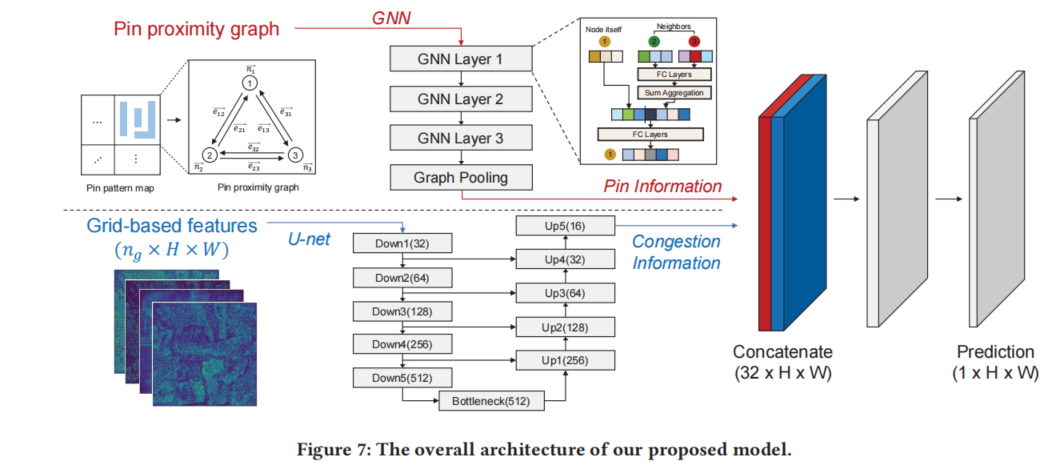
长/短RUDY：RUDY[17]是一个基于单元放置结果的快速路由拥塞估计器。它假设网线在其包围框内均匀分布。给定网为网格大小的边界框，定义的RUDY为边界框内通过单元网格的网的期望数目，即。此值被添加到与边界框重叠的每个网格中。我们生成两个不同的RUDY：长RUDY用于具有大边界框的网，短RUDY用于短网。如果一个网的包围框的半参数长度大于15格，则将其归类为长网。

水平/垂直布线能力：如果有许多金属轨道可用，即使在布线需求高的地区，也可以在不产生DRV的情况下成功进行布线。由于布线是在水平和垂直方向上进行的，我们使用每个网格在每个方向上的水平/垂直金属轨道的总数作为特征。

水平/垂直网密度：与RUDY类似，表示在水平/垂直方向上通过的网的预期数量，在[18]中首次提出。给定大小为网格的的边界框，每一行和每一列的单位网格的水平密度为，垂直密度为。这些值被添加到与边界框重叠的网格中。

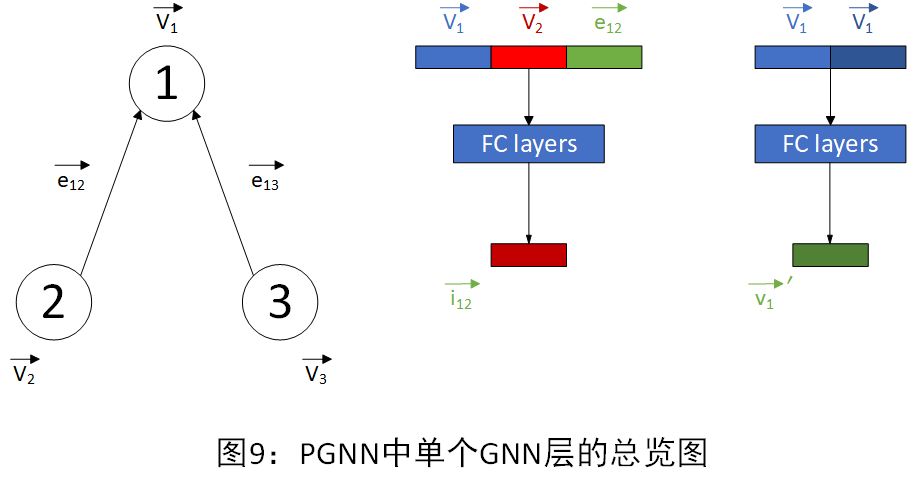
**模型建立**

PGNN（Pin and Global Routing Neural Network）是我们提出的模型，由三个模块组成：GNN模块用于建模网格内的引脚信息，U-Net模块用于建模路径拥挤情况，最终预测模块接受这两个模块的输出并生成预测结果。图7展示了我们的PGNN架构。GNN模块和U-Net模块适应不同类型的输入特征。GNN模块的输入特征是引脚接近性图，用于捕捉网格内的局部引脚可访问性。另一方面，U-Net模块以一组基于网格的特征作为输入，从整体的角度生成每个网格的表示，同时保留了全局的路径拥挤情况。然后，将两个模块的输出连接起来，最终的预测模块生成DRC（设计规则检查）热点预测图。如果预测值超过给定的阈值，我们将相应的网格报告为DRC热点。



由于我们的引脚接近性图具有边缘特征，传统的图神经网络（GNN）如GCN（图卷积网络）[10]和GAT（图注意力网络）[11]不能直接应用于我们的图中。因此，我们开发了一种新颖的图神经网络架构，非常适用于我们的图。

在我们的问题中，从GNN获得的节点特征表示相应节点代表的引脚的可访问性。我们的GNN层通过考虑相邻节点的信息来迭代更新节点特征。图8展示了在单个GNN层中更新图的节点特征的过程。



请注意，每个节点代表网格中的引脚。为了更新节点 的特征（例如，图8中的，GNN层首先计算，它表示访问引脚时相邻引脚的干扰。由于受到边缘特征和节点特征的影响，可以计算如下：



其中 是全连接层的可学习参数集合。所有相邻引脚的总干扰是所有单个干扰的总和，可以表示为：



其中是节点的相邻节点集合。最终，节点特征通过全连接层进行更新，更新过程是在将原始节点特征与总干扰进行拼接后进行的：



其中是全连接层的一组可学习参数。

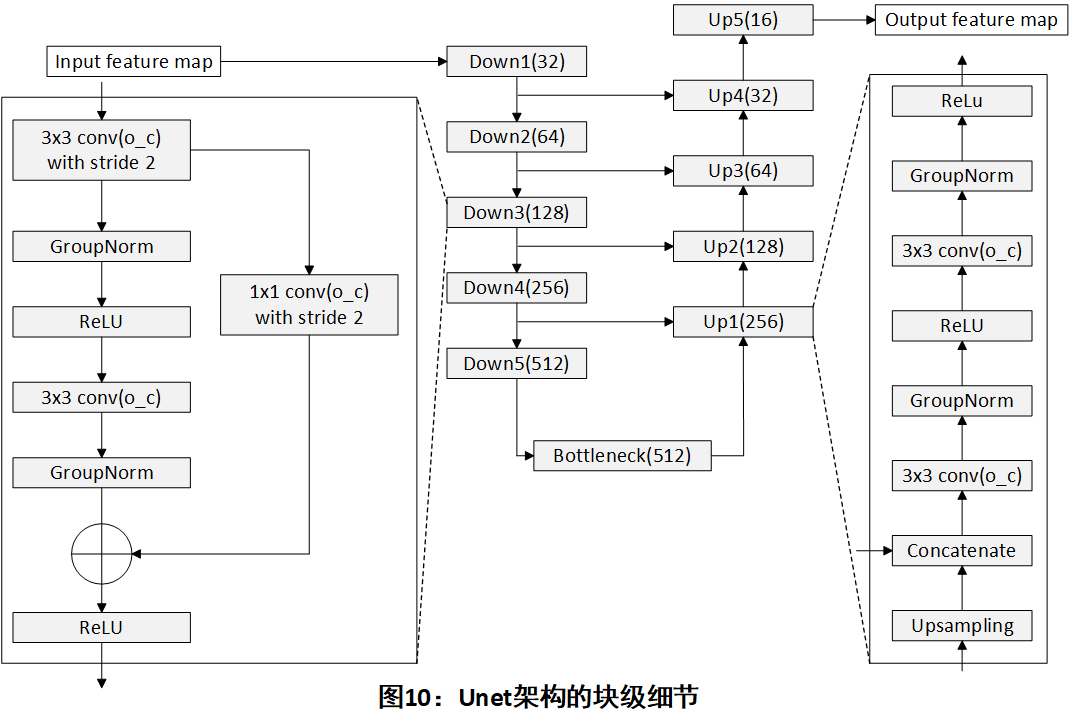
对于每个网格的引脚邻近图，通过处理三个连续的图神经网络（GNN）层来获得节点特征。为了高效地处理它们，我们将所有网格的图表达为一个单一的超图，以便可以应用并行处理。为了进行基于网格的预测，我们通过应用图平均池化将所有节点特征转换为全局图特征，这仅仅是取所有节点特征的平均值。然后，我们将为每个网格生成的图特征向量转换为三维矩阵以恢复空间信息。

**PGNN中的U-net体系结构**

图9显示了PGNN中U-Net的详细架构。图中每个块括号中的数字表示输出核的数量。U-Net的输入是在第3.3节中描述的基于网格的特征，大小为10×H×W，其中10是基于网格的特征的数量。对于U-Net的编码路径，我们采用了ResNet [19]骨干架构，它由五个采用快捷连接结构的下采样块组成。每个块对其输入应用一对3x3卷积层。第一个卷积层使用步长2来减半输入的大小。快捷连接使用步长2的1x1卷积来控制输出维度。因此，编码器将输入大小减小了32倍。

随后，图9中的瓶颈块通过3x3卷积层更新编码器的输出，然后解码器通过处理五个上采样块将压缩的特征图恢复到原始大小。每个上采样块使用转置卷积层将输入大小加倍，并将加倍后的输入与同一块级别（从瓶颈块开始计算）的下采样块产生的特征图进行连接，然后执行3x3卷积。我们在每个块中采用组归一化[20]和ReLU激活层。解码器的输出大小为16×H×W。

PGNN的最终预测模块将GNN和U-Net的输出特征图串联起来，得到32×H×W的特征大小。然后，该模块经过两个3x3卷积层处理，一个具有16个核，另一个具有1个核，生成大小为H×W的最终预测图。最后，应用sigmoid层将其转化为[0, 1]的概率分数。如果分数超过给定的阈值，则将相应的网格分类为DRC热点网格。

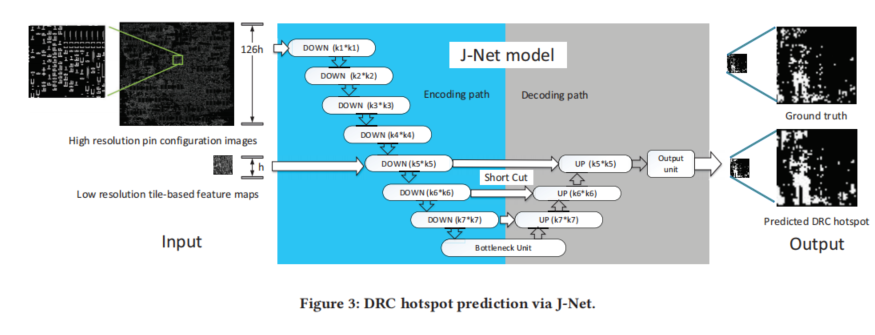


**PGNN中的最终预测**

我们的最终预测模块连接了GNN和U-net的输出特征图，得到的特征大小为32×H×W。然后，该模块处理两个3x3的卷积层，一个有16个内核，另一个有1个内核，以生成大小为H×W的最终预测图。最后，应用s型曲线层进行[0,1]概率评分。如果得分超过了给定的阈值，则将对应的网格划分为DRC热点网格。

**自定义J-Net卷积网络体系结构对10nm以下工艺节点的DRC热点预测**

模型架构： 论文中提出的模型称为J-Net，是一种定制的卷积神经网络架构，能够处理不同分辨率的输入特征。J-Net的设计强调了引脚可访问性，并且不需要全局布线信息。J-Net架构灵活，能够自动调整以适应不同设计的特征分辨率。J-Net基于编码器-解码器结构，通过多次下采样和上采样单元来处理特征，并通过调整卷积层的核心大小来适应不同分辨率的输入通道。



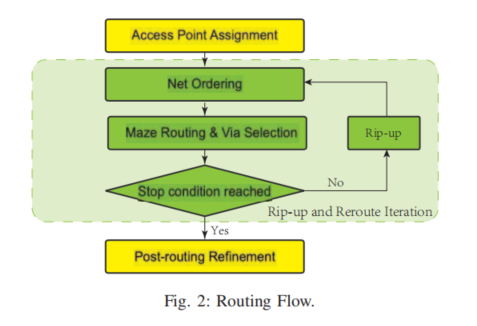
模型运行： J-Net的输入是一个单元放置解决方案，包括知识产权（IP）模块，输出是一个2D数组，每个条目对应一个瓦片，并指示每个瓦片是否是DRC热点。J-Net的原始输出是一个介于[-1, 1]之间的实数集合，通过设置阈值为0，正数表示热点，非正数表示非热点。J-Net在7纳米工艺节点的12个工业设计上进行了评估，结果显示，与最近三篇工作相比，J-Net在保持相似误报率的同时，分别提高了37%、40%和14%的真实阳性率。

此外，论文还介绍了J-Net的一些关键特点，如输入通道的不同分辨率处理、编码器和解码器级别的数量调整、卷积操作次数的减少，以及如何通过数据增强技术来处理训练数据量有限的问题。论文的实验部分详细描述了测试用例、数据收集、比较设置、训练和测试方案、性能评估指标以及实验结果。最后，论文总结了研究成果，并对未来的研究方向进行了展望。

1. **AI在网络排序（net order）中的应用**

**3.1 传统网络排序（net order）**

Net order在集成电路设计中扮演着重要的角色，对布线效果优化和资源利用起到关键作用。传统的net order方法通常基于启发式规则，如按照网络区域大小的降序排列。布线顺序对最终的布线质量具有重要影响，包括总线长、通过数量和设计规则检查（DRC）违例数量。通过合理的网络排序策略，可以优化布线效果，提高电路的性能和可靠性。布线是将逻辑电路映射到物理层的关键步骤，合理的网络排序能够减少信号路径的长度，降低延迟，并最小化电路中的信号干扰和噪声。此外，通过优化网络的布线顺序，可以更有效地利用布线资源，如通道和连线资源。合理的资源利用可以降低布线的复杂性和成本，并提高设计的可扩展性和可制造性。随着技术的发展，引入更多特征和机器学习方法来优化网络排序，可以进一步改进布线效果，减少设计规则检查违例和通过数量，提高布线质量和性能。因此，对于集成电路设计来说，传统的net order概念和相关方法的重要性不容忽视，它们为设计工程师提供了有力的工具来优化布线结果，确保电路的可靠运行和性能优越性。传统的布线流程如下图所示：



**3.2 AI在网络排序（net order）中的应用**

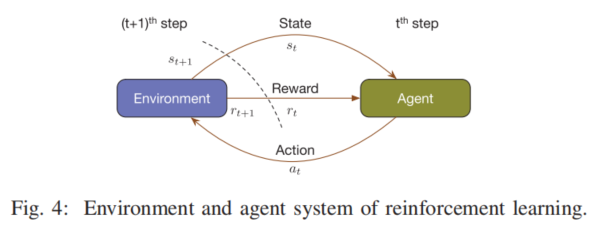
**3.2.1强化学习的概念**

强化学习（Reinforcement Learning, RL）是一种机器学习方法，通过与环境的交互学习最优策略。在RL中，代理根据观察到的环境状态选择动作，并根据获得的奖励信号来调整策略，以最大化累计奖励。关键组件包括状态空间、动作空间、奖励信号、策略和值函数。RL具有自动化策略学习的能力，适应性和泛化能力强，并且已经在多个领域得到应用。

在详细布线中，RL被用于优化网络排序策略。传统的网络排序方法往往基于启发式规则，而RL可以通过定义合适的状态空间、动作空间和奖励函数，自动学习并优化网络排序策略。实验结果显示，使用RL框架进行网络排序优化可以显著减少设计规则检查违例数量和总布线成本。这些实验结果表明，RL在解决复杂优化问题上具有潜力，并为布线设计提供了有效的解决方案。

强化学习是一种强大的机器学习方法，具有自动化策略学习和适应性强的特点。在详细布线等领域，RL的应用已经取得了一定的成功，并展示出其在优化问题中的潜力和优势。

强化学习使代理能够通过与环境的交互来学习策略，以最大化累积奖励。如**图4**所示，在每个步骤t，代理观察到一个状态，根据采取一个动作，接收到一个奖励，然后状态以随机方式转移到下一个状态。目标是学习一个策略π(a|s)，从任何状态s开始，最大化预期的累积奖励。



**3.2.2 问题定义**

**问题1（网络排序）。**给定一组网络N，训练一个网络排序策略，可以为每个网络∈N生成一个排序得分si，供顺序详细路由器使用。同时应优化以下指标：(1) 所有网络的总线长，(2) 使用的总过孔数量，(3) DRC违规数量。

**3.2.3 异步强化学习算法**

**基本的强化学习设置**

我们定义状态空间、动作空间和奖励如下：

状态空间 S：状态 s 是所有网络特征的集合表示。表I总结了每个网络的七个特征。第一个特征是其路由区域的大小。第二个特征是其度数，表示与其路由区域重叠的网络数量。第三个特征是到目前为止的路由/重新路由次数。其余四个特征是其成本信息，包括违规成本、线长、过孔数量和金属层分配。

**动作空间 A：**一个动作 a 是一个实数向量。每个数被定义为一个网络的排序得分。

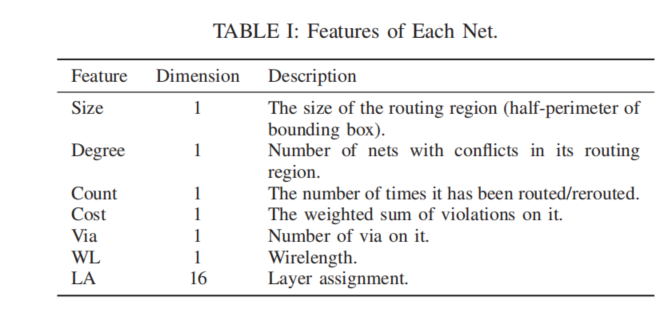
**奖励 R：**给定排序得分（动作 a），环境（路由器）将提供其反馈（即评估指标）。代理根据环境的反馈获得奖励。奖励 r 的定义如下：



其中和分别表示代理的动作a和Dr.CU默认策略所实现的所有网络的总成本。总成本C的定义如下：



其中xi|i ∈ {1, 2, 3, 4}是ISPD竞赛中使用的评估指标，包括短违规、间距违规、过孔数量和线长，wi|i ∈ {1, 2, 3, 4}是上述指标的权重。代理的目标是学习一种策略，以最大化奖励。

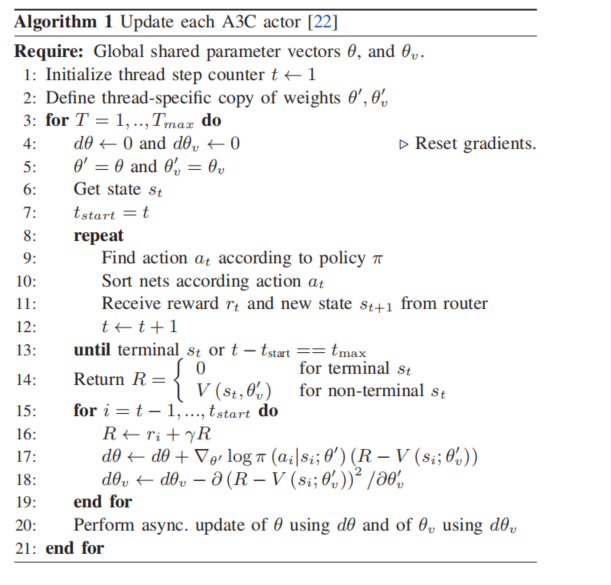


**A3C框架**

昂贵的环境查询是强化学习中的一个典型挑战，导致收敛速度较慢且训练时间过长。为了应对这个问题，我们采用了一种异步优势actor-critic（A3C）方法[22]，其中有多个并行运行的actor-critic代理。如图5所示，每个代理都有策略和值网络的本地副本。它在环境中执行动作以探索解空间，并使用不同的策略。不同的代理在训练过程中异步地更新全局网络。

A3C维护一个策略π(|;)和值函数的估计V(;)，其中θ和是全局共享的参数向量。在每个个动作或达到终止状态时，策略和值函数进行更新。

算法1展示了每个actor如何进行更新。初始化之后，每个代理获取全局共享网络的副本，具有参数和（第5行），然后运行策略步或直到达到终止状态。最后，代理在其进程中计算梯度（第17-18行），然后异步地更新全局共享网络。



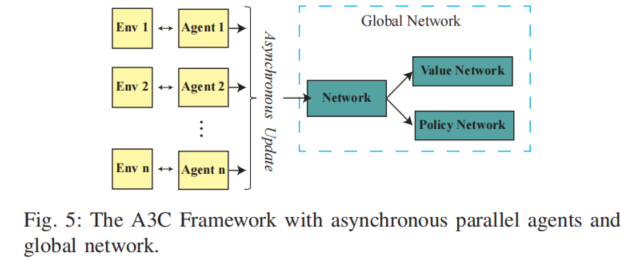
**A3C（Asynchronous Advantage Actor-Critic）**是一种异步并行的强化学习算法，通过多个并行的actor-critic代理来提升学习效率和加速收敛。每个代理都有自己的策略网络和价值网络副本，并在不同的环境中独立进行探索。通过异步更新全局网络，A3C能够更快地收敛并减少总体训练时间。

A3C的主要特点包括异步并行训练、策略网络和价值网络的结构以及网络架构的独立性。多个代理并行运行，每个代理在自己的环境中执行操作，并探索解空间。策略网络输出动作的概率分布，而价值网络评估特定状态下的预期回报。通过独立的网络架构，策略网络可以在不同设计中共享，扩展了A3C的应用范围。

A3C算法的流程包括初始化、本地副本和探索与更新。全局共享参数向量被初始化，每个代理获取全局网络的本地副本，并在其环境中执行一定步骤或达到终止状态。代理计算梯度并异步更新全局共享网络。

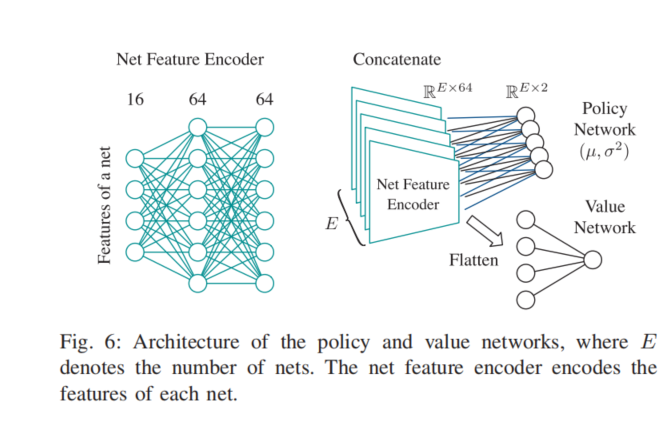
A3C具有提高训练效率、减少训练时间和适应性强的优势。通过异步并行训练，A3C能够加速策略优化和收敛过程，提高学习效率。同时，全局网络的异步更新能够更快地收敛，减少总体训练时间。另外，A3C的策略网络设计可以在不同的任务中共享，具有广泛的应用性。

A3C框架在强化学习中具有重要的地位，通过异步并行的训练方式和独立的网络架构，提高了学习效率和训练速度。它在各种任务中展示了卓越的性能，并在详细布线等领域取得了显著的成果。



**网络架构。**在A3C框架中，我们需要两个模型，一个策略网络和一个价值网络。策略网络接收状态s并输出两个数组（，），表示动作的概率分布。我们通过从该概率分布中进行采样来选择动作。我们用π(a|s)表示在给定状态s的情况下采样动作a的概率。价值网络输出值函数V(s)（状态s和动作a的预期回报），用于确定在特定状态下的优势程度。直观地说，策略网络告诉我们网络的排序得分，而价值网络根据未来的奖励评估这些得分。

图6显示了这两个模型的网络架构。我们以一种特殊的方式设计模型，使得策略模型可以用于具有不同数量网络的不同设计中。为了将网络架构与设计中的网络数量解耦，我们引入了一个逐个网络特征编码网络，用于独立地编码每个网络的特征。然后，我们将编码后的特征连接到策略网络和价值网络中。例如，给定一个具有E个网络的设计，编码器将把RE×16的输入特征张量编码成一个RE×64的张量。策略网络接收该张量并为所有网络生成一个排序得分数组，即RE×2（每个网络的概率分布的均值和方差）。然后，我们对每个网络从正态分布中进行采样，以获得其排序得分。在我们的实现中，μ由线性层建模，由softplus层建模。价值网络将特征张量展平，并通过具有E×64隐藏单元的全连接层得到一个标量输出。

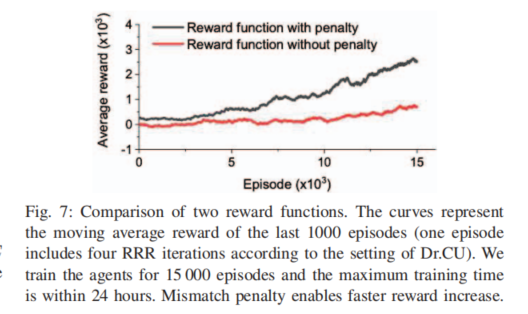


这种网络架构的主要优点是，策略网络可以在不同的设计中共享，因为我们实际上是对每个网络的排序得分进行逐个建模，仅依赖于其特征。虽然使用一个更复杂的模型来关联多个网络的特征可能有助于探索更好的策略，但当前的架构仍具有足够的表达能力来验证在解决网络排序问题中使用强化学习的主要思想。我们将在未来探索更复杂模型的可能性。

**惩罚函数。**通常的强化学习框架会以随机方式初始化神经网络，这可能导致在我们的问题中收敛速度较慢，特别是当从环境中获取奖励（即运行路由器）非常耗时时。另一方面，我们对这个问题有先验知识，即在Dr.CU中使用路由区域大小的默认排序策略相比随机策略来说是一个通常较好的策略。将这种知识纳入到训练中有可能加速训练过程。因此，方程（1）被修改为：

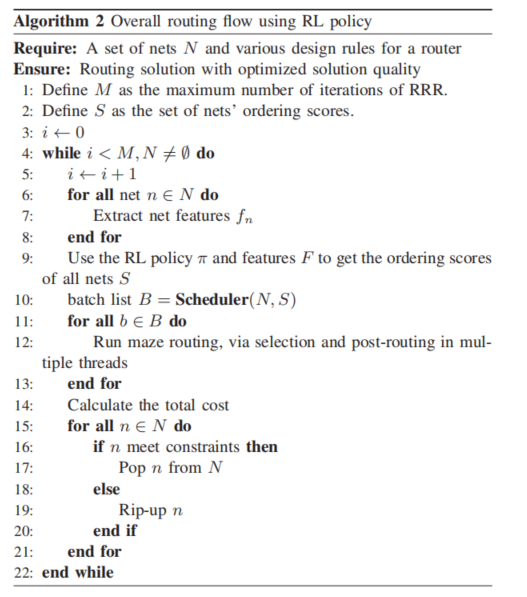


其中是预测的排序得分与路由区域大小之间的差异，α是用户定义的参数，k是要进行路由的网络数量。参数α仅在训练的早期阶段为正，并在之后设为零。图7比较了两种定义奖励函数的方法的学习速度。结果表明，添加不匹配惩罚的方法往往学习更快。由于我们只在训练的早期阶段应用不匹配惩罚，它将加快训练速度，但不会限制探索空间只局限于Dr.CU中使用的启发式排序策略。



**整体路由流程**

一旦得到策略，整体路由流程如算法2所示。在获得排序得分后（第9行），我们利用Dr.CU来完成每个RRR迭代（第10-19行）。具体来说，在RRR迭代的开始时，我们通过根据得分对网络进行排序并将它们分成批次来安排所有批次（第10行），使得批次内的网络不相互冲突，并且可以并行路由以减少运行时间[16]。如果未满足RRR停止准则，迭代将继续进行，直到达到最大迭代次数。



1. **基于注意力机制的Track Assignment**

**4.1传统轨道分配**

轨道分配是指在集成电路设计中，将电路中的信号线路（iroutes）分配到指定的轨道（tracks）上的过程。它是现代VLSI设计流程中的一个关键步骤，用于解决电路布局中的路由性能和可行性问题。

轨道分配的目标是通过将信号线路分配到适当的轨道上，最小化不同信号线路之间的重叠，从而提高整体路由的质量和可行性。通过减少不同信号线之间的干扰和冲突，轨道分配可以提高电路的稳定性、可靠性和性能。

轨道分配通常是在全局路由之后进行的，全局路由将信号线路连接到全局网格中的全局单元（g-cells）。在轨道分配过程中，每个信号线路被分配到一个轨道上，并且需要满足一些约束条件，如最小间距、相位等。

为了更准确地估计轨道的可行性和解决详细布局中的问题，一些方法使用基于协商的算法来进行轨道分配。这些算法会考虑全局路由提取的信号线路信息以及局部网络的布局。通过协商不同信号线路之间的重叠情况，可以更准确地判断哪些区域存在重要的路由问题，并进行相应的优化和修复。

轨道分配的目标是生成一个满足约束条件的、最小化重叠的轨道分配方案。这需要在保证全局连接性和信号完整性的前提下，尽量减少冲突和干扰。轨道分配的算法和方法可以根据具体的设计需求和约束进行调整和优化，以提高布局的质量和性能。

**4.2基于学习的轨迹分配方法**

**使用带有监督的基于注意力的策略模型的跟踪分配详细路由**

主要介绍了在先进节点模拟电路中解决轨道分配详细布线问题的一种基于机器学习的方法。提出了一种基于注意力机制的强化学习（RL）策略模型，并通过引入监督学习的方式利用传统遗传算法（GA）生成的解来改进该模型。通过最小化RL策略模型输出与遗传算法得到的解分布之间的Kullback-Leibler散度损失，该方法能够在离线环境中通过监督学习训练路由器，并且将运行时间性能提高近100倍。该方法生成的解质量与遗传算法生成的解相匹配，并且在复杂问题上表现出良好的解质量，与传统的基于注意力的RL方法相比并不牺牲运行时间性能。该方法的能力使得路由器能够从示例设计中学习，并训练出与专家生成的类似解，从而显著影响设计流程，提高设计探索和路由驱动布局的能力。

**4.3.模型建立**

为了简化设计规则的应用，本章节采用了Width Spacing Patterns（WSP）作为设计规则的抽象表示。WSP包含了不同金属导线的宽度和间距参数。通过将路径限制在WSP的行和轨道上，可以使用修改后的加权二分匹配公式预先施加许多与全定制设计相关的设计规则。

本章提出三种方法，即遗传算法（GA）、基于注意力机制的强化学习（ARL）和新的监督强化学习（SRL），都使用了相同的基本方法。这些方法的目标是确定引脚对的顺序，以便在使用模式路由器进行布线时最大化解决方案的质量，其中解决方案的质量通过完成的路径数量和总线长度来衡量。WSP的使用使得这三种方法可以在相同的基础上进行比较，并且可以将适用的设计规则抽象成一种统一的形式。

**基于注意力的强化学习策略模型**

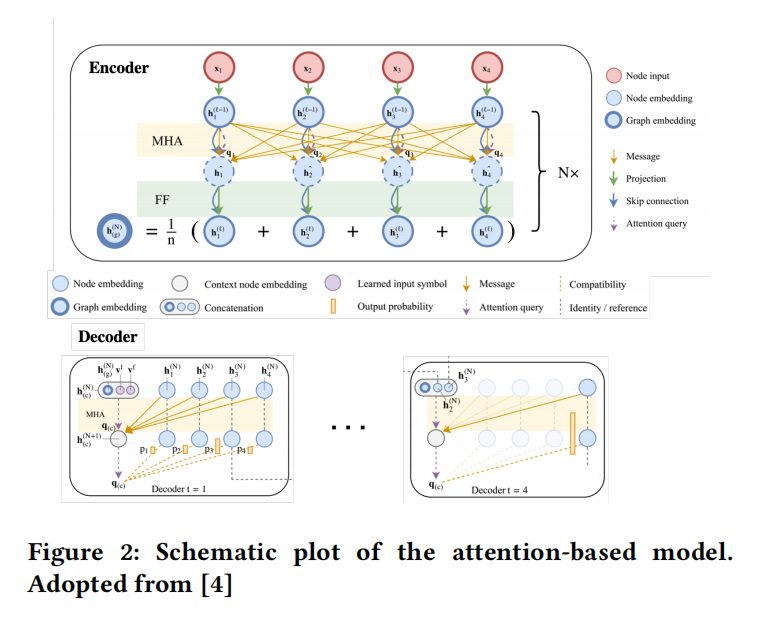
2.2基于注意力的强化学习策略模型最近的研究表明，强化学习在组合优化问题上取得了成功，与基于启发式方法或精确方法相比，它具有更好的泛化能力和运行时性能[1,2,7]。在这些作品中，Kool等。[4]使用基于注意的强化学习策略模型来解决旅行商问题(TSP)。与以前的方法相比，它们具有推广到未见问题的能力，从而实现了最先进的性能[12,15]。在用策略模型求解TSP时，一个解可以定义为一个行程n，它由节点(城市)的有序序列组成。在每一步中，策略模型在接下来可能访问的节点上生成一个概率分布。TSP问题的解可以在n步中得到，其中n表示TSP问题中的城市数量。基于这种机制，该工作通过将每一步的概率分布相乘来定义问题策略，如公式1所示。



为了获得稳健的策略模型，需要对策略模型的参数进行优化。基于问题策略和强化中使用的策略梯度定理，策略模型的损失函数梯度可表示为:。这可以根据策略梯度定理[14]在公式2中计算得到，它为计算难以处理的梯度信息以优化模型参数提供了一种可行的方法:



应用策略模型的结构是基于注意的编码器-解码器(图2)。该模型是Graph Attention Networks的一种变体，其优势是能够以序列独立的方式解决序列问题。



受到模拟详细布线问题和TSP之间的相似性的启发，先前的研究[8]成功地将基于注意力机制的强化学习（ARL）策略模型应用于解决轨道分配详细布线问题。在那项研究中，ARL能够以一种可泛化的方式解决详细布线问题。ARL最重要的贡献是提出了一种从网表中提取引脚对的新型快速排序方法。如果没有适当的排序，详细布线通常需要使用计算量大且次优的撕除和重布线策略来解决[3]。ARL是在没有任何监督的情况下进行训练的。

本研究基于ARL的基础上引入了遗传算法（GA）解决方案提供的监督信息。我们将得到的方法称为有监督强化学习（SRL）用于布线，并在接下来进行讨论。该研究受到最近关于将强化学习与监督学习相结合的工作的启发[13, 16]。

**经过监督训练的政策模型**

为了对策略模型进行监督训练，需要生成一个类似于问题策略输出的目标（标签）。我们使用遗传算法（GA）的解作为标记数据。我们将这些在种群中获得的解转化为概率分布，以供策略SRL模型使用。具体而言，考虑一个给定的问题实例s，其中包含n个需要使用模式路由器进行排序和布线的双引脚对。在应用GA求解器之后，可以获得一组经过优化（但不一定是最优的）的双引脚对序列。在序列的每个位置，t = 1，...，n处，可以获得双引脚对之间的经验分布。基于这个分布，通过对，t = 1，...，n，进行乘法运算，可以得到问题实例s的策略标签分布。为了方便进行代数运算，我们使用log-label的形式：



和 都是向量，并且对向量中的各个分量应用了对数运算，因此  也是一个向量。根据注意力基础策略模型给出的标签和问题策略，基于Kullback-Leibler（KL）散度制定了损失函数。



这种损失编码了两个分布之间的交叉熵损失；一个来自策略模型，另一个来自遗传算法解。我们使用了一个离散化的版本：



**问题表述**

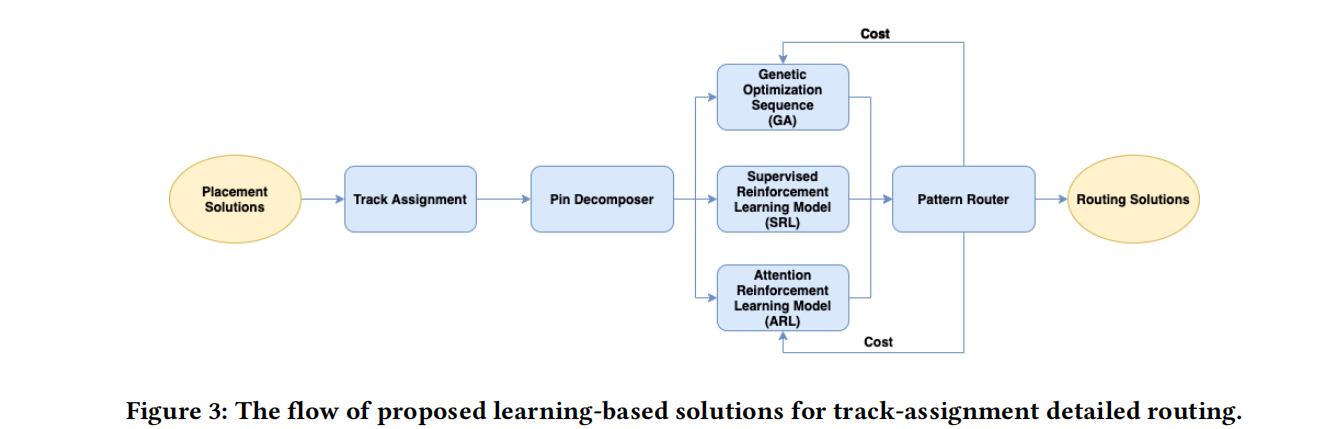
图3展示了针对详细布线任务的提出的基于学习型解决方案流程。流程的输入是一组布线解决方案，这些解决方案针对给定的模拟电路设计的子区域进行布线。子区域的布线解决方案指定了设备的位置和包含需要在详细布线阶段连接的设备分组信息的网络列表。假设给定模拟设计的特定区域的布线解决方案集S遵循某种分布(即不同设备在设计空间中不同位置出现的概率)，而不是随机分布。这是基于这样的观察:布线解决方案通常是由考虑类似目标(如设备密度或潜在下降)生成的。在轨道分配和引脚分解之后，每个布线解决方案的详细布线被制定为连接一组两引脚对，这些引脚对具有预定的空间位置，并遵循序列。可能的序列数量由两引脚对集N内的所有排列决定。顺序详细布线问题正式表述如下:

顺序详细布线:考虑一组在详细布线区域表示为图G(V,E)上的两引脚对，其中引脚分布在不同的顶点和边上。只有水平和垂直方向是可行的布线方向，布线容量为1。顺序详细布线引擎分两步解决详细布线问题。首先，通过求解器生成优化后的两引脚对序列π。然后，按照序列元，)模式路由器顺序地布线每个两引脚对，同时共享相同的图G(V，E)。

目标函数:要优化的指标包括:(1)所有两引脚对路径的总线长，(2)未布线两引脚对的总数，即开口数。当两引脚对无法布线时，开口数增加一个。为了便于实现，使用开口总数和总线长的加权和作为目标函数，即成本，以最小化:



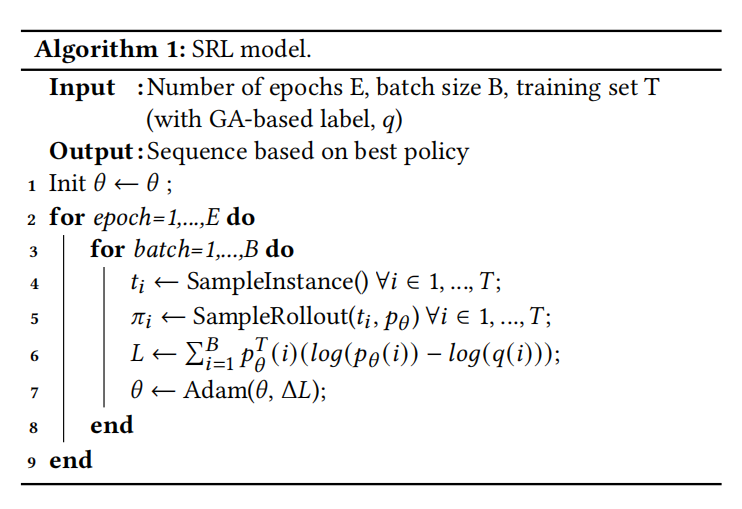
成本越低，解决方案的质量就越好。最终用户可以通过调整两个项的权重来不同地偏重目标函数的两个组成部分。



**算法**

如图3所示，我们分析了遗传算法(GA)、自适应重标定学习(ARL)和监督重标定学习(SRL)方法。

这三种方法都使用相同的轨道分配方法和引脚分解算法(Kruskal算法[5])。所有三种方法都使用相同的模式路由器，包括"L"和"Z"形状的布线，这为比较这些算法提供了一个共同的基础。基于GA的排序通过多代迭代优化给定问题中的序列π。它使用基于排列的交叉和变异操作来创建后代。ARL被用作一种通过在以前工作中的一组问题上进行自我训练来学习生成优化序列的方法，这是建立在基于注意力的强化学习策略模型之上的。在这项工作中，我们提出了ARL的监督版本，我们称之为SRL，它基于KL散度损失和GA解决方案作为监督，如算法1所示。在训练期间，给定设计的60%的布线解决方案用作训练数据，20%用作验证，20%用作测试。给定训练问题，GA首先在10代后根据公式Egn.3生成以分布q形式的结果。从GA获得q后，通过计算策略模型输出的分布p0和q分布之间的KL散度损失来启动SRL训练。然后，损失用于通过反向传播更新策略模型参数，我们执行100个周期。一旦训练完成，SRL以前向方式为同一数据集中以前未见过的问题(布线)提供引脚对序列。训练后的SRL(以及ARL)的运行时计算成本可以忽略不计，与GA相比，因为在运行时只涉及矩阵乘法。



**模型评估**

在评估遗传算法GA、自适应重标定学习ARL和监督重标定学习SRL这三种方法的性能时，研究者们采用了两种模拟设计问题（biasamp和sarfsm）并创建了不同规模的数据集。所有算法均在相同参数设置下使用Python 3.6和Pytorch框架在无GPU的工作站上实现和训练。评估主要通过训练时间、成本相关性分析、成本与运行时间的比较、成本偏差的直方图、SRL与ARL之间的相关性分析以及解决方案的相似性对比来进行。此外，还计划进行更大规模数据集的统计分析以验证观察到的趋势。研究结果显示，SRL在解决方案质量上略低于GA，但具有显著更优的运行时间，通常在1%的GA运行时间内生成解决方案，而ARL的性能与SRL相当。**参考文献**

[1] Thomas D Barrett, William R Clements, Jakob N Foerster, and Alex I Lvovsky. 2019. Exploratory combinatorial optimization with reinforcement learning. arXiv preprint arXiv:1909.04063 (2019).

[2] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. 2016. Neural combinatorial optimization with reinforcement learning. arXiv preprint arXiv:1611.09940 (2016).

[3] William A Dees and Patrick G Karger. 1982. Automated rip-up and reroute techniques. In 19th Design Automation Conference. IEEE, 432–439.

[4] Wouter Kool, Herke Van Hoof, and Max Welling. 2018. Attention, learn to solve routing problems! arXiv preprint arXiv:1803.08475 (2018).

[5] Joseph B Kruskal. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical society 7, 1 (1956), 48–50.

[6] Koen Lampaert, Georges Gielen, and Willy Sansen. 1996. Analog routing for manufacturability. In Proceedings of Custom Integrated Circuits Conference. IEEE, 175–178.

[7] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. 2018. Combinatorial optimization with graph convolutional networks and guided tree search. In Advances in Neural Information Processing Systems. 539–548.

[8] Haiguang Liao, Qingyi Dong, Xuliang Dong, Wentai Zhang, Wangyang Zhang, Weiyi Qi, Elias Fallon, and Levent Burak Kara. 2020. Attention Routing: trackassignment detailed routing using attention-based reinforcement learning. arXiv preprint arXiv:2004.09473 (2020).

[9] Enrico Malavasi and Alberto Sangiovanni-Vincentelli. 1993. Area routing for analog layout. IEEE transactions on computer-aided design of integrated circuits and systems 12, 8 (1993), 1186–1197.

[10] Prasanth Mangalagiri. 2019. Analog Layout Synthesis: Are We There Yet?. In Proceedings of the 2019 International Symposium on Physical Design. 127–127.

[11] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Sungmin Bae, et al. 2020. Chip Placement with Deep Reinforcement Learning. arXiv preprint arXiv:2004.10746 (2020).

[12] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takác. 2018. Reinforcement learning for solving the vehicle routing problem. In Advances in Neural Information Processing Systems. 9839–9849.

[13] Evan Shelhamer, Parsa Mahmoudieh, Max Argus, and Trevor Darrell. 2016. Loss is its own reward: Self-supervision for reinforcement learning. arXiv preprint arXiv:1612.07307 (2016). [14] Richard S Sutton and Andrew G Barto. 2018. Reinforcement learning: An introduction. MIT press.

[15] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In Advances in neural information processing systems. 2692–2700.

[16] Jin Zhang, Jiansheng Chen, Yiqing Huang, Weitao Wan, and Tianpeng Li. 2018. Applying Online Expert Supervision in Deep Actor-Critic Reinforcement Learning. In Chinese Conference on Pattern Recognition and Computer Vision (PRCV). Springer, 469–478.

[17] Keren Zhu, Mingjie Liu, Yibo Lin, Biying Xu, Shaolan Li, Xiyuan Tang, Nan Sun, and David Z Pan. 2019. Geniusroute: A new analog routing paradigm using generative neural network guidance. In Proc. ICCAD.

1. **主动学习在详细布线中的应用**

主动学习是一种适合处理大量未标记数据和昂贵标注成本问题的机器学习技术，为节省标注成本而不降低训练模型的性能提供了巨大机会。图2简要介绍了将主动学习应用于解决分类问题的概念。红色和绿色实心圆分别代表类别A和类别B的已标记数据，红色和绿色空心圆分别代表类别A和类别B的未标记数据。主动学习的目标是查询较少但更重要的数据的标签，以训练一个分类边界，从而对已标记和未标记的数据集都具有良好的性能。

主动学习的程序由两个阶段组成：(1) 迭代地采样和标注一些查询数据，(2) 根据查询的数据重新训练模型。在每次迭代中，它查询一个或一批样本进行标注，这些样本预计将能够提高当前分类模型的性能。有两种查询策略，即信息性和代表性，它们在先前关于主动学习的研究中被广泛使用。信息性被视为一个样本提高分类模型准确性或减少整体损失的能力，并在下一次迭代的训练中实现较少的不确定性。代表性描述了一个样本与潜在未标记数据之间的相关性。

问题定义

2.2 问题表述

本文的目标是训练一个基于单元库的引脚可访问性预测模型。顾名思义，我们方法的输入是一组单元库，记为 ，其中 m 表示单元库的数量。设 是中的标准单元集合，其中 n 代表中标准单元的数量。请注意，每个单元库中的标准单元数量可能不同。有了这些带有相应标准单元的单元库，我们可以如下表述问题:

问题1:给定一组单元库L，任务是开发一个基于主动学习的培训流程，以训练一个引脚可访问性预测模型，该模型可以应用于预测由于引脚可访问性差而导致的设计中的DRV发生。

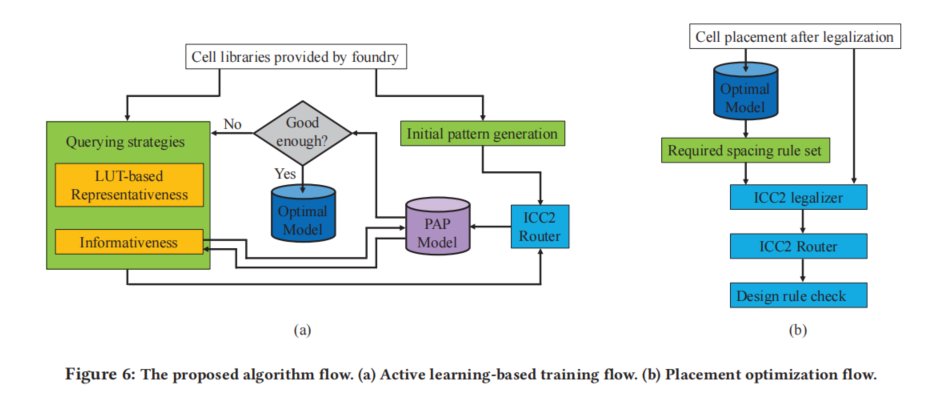
**数据提取**

**和上面引脚访问小节类似，主要讲解基于主动学习的模型训练流程。**

**算法流程**

所提出方法的概览在图6中给出，它由两部分组成：(1) 基于主动学习的模型训练流程和(2) 放置优化流程。图6(a)展示了所提出的基于主动学习的模型训练流程，该流程可以分为两个阶段：初始模式生成和查询策略。由于标准单元库是由代工厂或设计公司提供的，我们首先在初始模式生成阶段随机生成一些单元组合。然后，生成的单元组合被输入到ICC2路由器中查询标签，所有已标记的单元组合被用来训练初始的PAP模型。之后，我们将进入查询策略阶段，这需要基于查找表（LUT）的代表性评估和信息性计算。通过这两种定量分析，我们可以获得一组关键且未标记的单元组合。最后，所有标记的单元组合将被输入到PAP模型中进行重新训练，直到模型的性能足够好。

通过所提出的模型训练流程获得最优的PAP模型后，我们进一步应用该模型指导ICC2的合法化器优化单元放置。图6(b)展示了放置优化流程。输入是一个合法化的放置结果，首先采用训练好的模型计算放置单元之间的所需放置间距规则。然后，将所需的放置间距规则集成到合法化器中，以避免在后合法化阶段生成不良的单元组合。最后，进行路由和设计规则检查（DRC），以检查后合法化放置结果的质量。

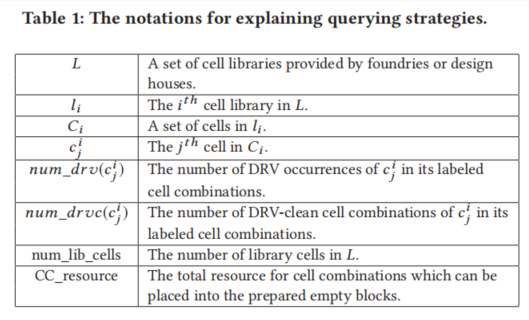


**初始单元组合生成。**

在开始迭代的基于主动学习的流程之前，应该准备好初始模型。由于我们没有任何信息来智能选择给定单元库中的好样本，我们只能从给定的单元库中随机组合一批单元组合。任务是训练一个PAP模型，该模型可以预测给定单元库中每个单元的引脚可访问性，同时考虑相邻单元。为了让初始模型充分考虑所有单元，每种类型的库单元被视为目标单元，并放置在单元组合的中心位置进行引脚可访问性评估。然后，我们随机选择两个具有随机方向的库单元，不违反PG轨道违规，分别紧邻目标单元的左侧和右侧。通过网络列表构建，我们将所有单元组合放置在准备好的空设计中，并调用ICC2路由器[32]来完成路由和DRC检查，以派生标签。由于所有单元组合都已标记，我们使用所有单元组合及其标签来训练初始模型。

**查询策略。**

尽管初始模型考虑了所有单元的引脚可访问性，但由于每个单元只考虑了少数邻接组合，模型的准确性较低。基于预训练的初始模型，需要更多的单元组合标签来增强模型。由于查询天文数字的单元组合成本过高，我们提出了两种查询策略来智能查询新的未见样本：(1)基于查找表（LUT）的代表性评估，以及(2)信息性计算，下面将详细说明。



基于LUT的代表性评估。

在这一部分中，我们试图挑选出更多具有更高DRV发生概率的代表性库单元，并确定每个库单元的查询次数。在详细说明代表性查询策略之前，一些符号在表1中定义。初始模型训练完成后，我们将拥有训练单元组合的标签。因此，我们可以创建一个查找表（LUT）来存储每个库单元的DRV发生历史，其中DRV发生历史用来记录与每个库单元相关联的有和没有DRV的单元组合的数量。有了每个库单元的DRV发生历史，我们可以直观地计算每个库单元的DRV发生概率，如下所示：



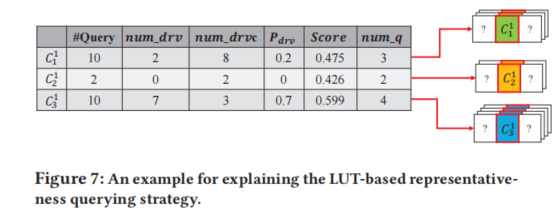
其中， 表示对 （即第 个库单元的第 个变体）发生DRV的概率估计。需要再次强调的是，PAP模型的目标是识别那些由于引脚可访问性差而发生DRV的单元组合。因此，我们的目标是查询那些具有更高DRV发生概率的库单元的更多单元组合。然而，对于那些在初始训练期间采样的单元组合都是无DRV的单元，我们不能立即筛选出这些单元。这是因为当这些单元与初始模型中未考虑的其他邻近单元相邻时，它们仍然有机会遭受DRV。因此，我们通过从当前概率中取零均值来计算每个库单元的代表性得分，然后应用函数以确保得分在[0,1]范围内，并且对于那些在初始训练中没有DRV发生的单元，得分不会为零。得分函数的公式如下：



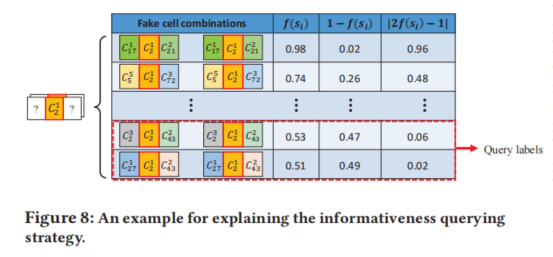
在这里，是一个用户定义的参数，越大，函数的曲线就越平滑。在获得所有库单元的代表性得分之后，我们可以按照以下方式计算每个库单元所需的查询次数：



请注意，是一个固定数字，它取决于可用内存的使用量。拥有更多的机器和每台机器上更多的内存，我们就可以为ICC2创建更多的空白块来路由未标记的单元组合。方程（4）中的floor函数用于防止未标记单元组合的资源不足。图7给出了一个简单的示例，用以解释基于LUT的代表性查询策略。请注意，在此示例中，α和分别被设置为1和10。根据DRV历史LUT，我们可以很容易地找到每个库单元的和的数量。因此，术语、和可以分别通过方程（1）、（2）和（4）计算得出。通过计算每个库单元的，我们可以得到每次迭代中每个库单元的确切查询次数，如图7最右列所示。



**信息性**。在确定了每个库单元的查询次数之后，我们的目标是为每个库单元挑选出最具信息性的单元组合，这些组合能够最大程度地提升模型性能。也就是说，我们关注那些模型没有足够信心的单元组合，因此查询这些组合的标签是可取的，并且能够提高DRV预测的准确性。我们以图7中的为例来解释信息性查询策略，如图8所示。所需的查询次数已经通过基于LUT的代表性评估获得。在信息性计算中，我们首先随机创建一批未见过的单元组合。然后，将所有单元组合输入模型以评估DRV预测得分，其中表示第i个单元组合，f代表用于评估DRV发生概率的假设函数[5]。计算完所有DRV概率后，可以通过轻松计算DRV清洁概率。最后，选择那些其DRV概率和DRV清洁概率之间差异最小的单元组合作为查询其标签的候选。如图8所示，对于，最后两个单元组合由于和之间差异最小而被查询。



**6.总结**

在电子设计自动化（EDA）领域，详细布线是一个至关重要的步骤，它直接影响到集成电路（IC）的性能、功耗和产量。随着半导体技术节点的不断进步，标准单元的尺寸不断缩小，导致标准单元I/O引脚的密度显著增加，这使得引脚的可访问性（pin accessibility）成为一个日益突出的问题。为了解决这一挑战，研究人员开始探索利用人工智能（AI）技术，特别是机器学习（ML）方法，以提高详细布线的效率和质量。

近年来，AI在EDA领域的应用取得了显著进展。例如，Pin Accessibility and Routing Congestion Aware DRC Hotspot Prediction using Graph Neural Network and U-Net这篇论文提出了一种基于图神经网络（GNN）和U-Net的新型机器学习模型（PGNN），用于预测详细布线阶段的设计规则检查（DRC）热点。该模型能够同时考虑引脚可访问性和布线拥塞对DRC热点的复合影响，通过实验验证，PGNN在所有基准设计上的性能均优于现有ML模型，实现了平均7.8%至12.5%的F1分数提升，同时推理时间比现有技术快5.5倍。

另一篇论文，DRC Hotspot Prediction at Sub-10nm Process Nodes Using Customized Convolutional Network，介绍了一种名为J-Net的定制卷积神经网络技术，用于在7纳米技术节点上对12个工业设计进行DRC热点预测。J-Net架构灵活，能够处理各种输入和输出分辨率要求，且无需使用全局布线信息，其在真实阳性率（TPR）上相较于其他三篇近期工作分别提高了37%、40%和14%。

此外，Pin Accessibility Prediction and Optimization with Deep Learning-based Pin Pattern Recognition论文提出了一种基于卷积神经网络（CNN）的模型，使用引脚模式作为主要特征来预测由于引脚可访问性差而可能引起的设计规则违规（DRV）。该方法不仅能够精确预测M2短路DRVs，还能够在物理设计期间优化引脚可访问性，显著减少了M2短路和总体DRVs的数量。

在Lookahead Placement Optimization with Cell Library-based Pin Accessibility Prediction via Active Learning论文中，作者提出了一种基于主动学习技术的单元库引脚可访问性预测方法。这种方法不需要大量的路由设计预先生成，而是直接从给定的单元库中获取训练数据，通过主动学习查询策略，模型能够迭代地改进性能，并且适用于所有引用相同单元库集的设计。

最后，Enhanced pin-access prediction and design optimization with machine learning integration论文提出了一种集成机器学习算法的引脚可访问性检查和优化方法，该方法通过使用机器学习算法来提高引脚可访问性预测的准确性并减少DRV计数。实验结果表明，使用该方法可以减少47%的DRV计数，同时运行时间增加了23%。

这些研究表明，AI技术，特别是ML算法，在EDA领域的详细布线中具有巨大的潜力。通过精确预测引脚可访问性问题和优化布线策略，AI不仅能够提高布线的质量，还能够显著减少设计迭代所需的时间，从而加速整个芯片设计流程。随着技术的不断发展，我们可以预见，AI将在EDA领域扮演越来越重要的角色，为芯片设计带来更多创新和突破。