项目指导规范：基于 Graph Diffusion 与 Graph U-Net 的 Macro 布局规划 (Floorplanning)

1. 项目概述 (Project Overview)

本项目旨在利用**生成式 AI (Generative AI)** 的前沿技术，创新性地解决 VLSI 物理设计流程中最关键且耗时的 **Macro Floorplanning (宏单元布局)** 问题。

核心目标是利用 **Diffusion Model (扩散模型)** 强大的分布建模和解空间探索能力，生成不仅满足物理约束（如无重叠）且具有优异 **对齐度 (Alignment)** 的布局方案。这对于后续的电源网络规划、拥塞控制以及标准单元放置至关重要。

本方案的根本创新点在于彻底摒弃传统的基于像素网格 (Pixel-based) 的图像处理方法。传统的像素方法在处理大规模芯片布局时，面临网格稀疏性高、分辨率受限以及计算资源浪费在大量空白区域等问题。相比之下，本项目转而采用 **基于图 (Graph-based)** 的方法，利用 **Graph U-Net** 直接在 **Channel Graph (通道图)** 的非欧几里得拓扑结构上进行去噪和坐标生成，从而能够更精准、高效地捕捉 Macro 之间的相对位置关系和逻辑连接。

2. 问题定义 (Problem Definition)

- **输入 (Input)**:
  - **Macro 几何信息**: 一组异构（不同尺寸、不同纵横比）的 Macro 集合 $M = \{m_1, m_2, ..., m_N\}$。对于每个 Macro m_i，模型接收其固定的宽度 w_i 和高度 h_i 作为不可变的条件输入。
  - **拓扑结构**: 初始的连接关系或基于简单启发式算法（如 B*-tree 或 Sequence Pair）生成的初步布局，用于构建初始的 **Channel Graph**。该图结构显式地表达了 Macro 之间的相邻通道关系。
  - **逻辑连接 (Netlist)**: 详细的网表信息，定义了 Macro 之间以及 Macro 与 I/O 端口之间的连接权重（Connectivity Weights）。这是优化线长的基础数据。
- **输出 (Output)**:
  - 布局方案中每个 Macro m_i 的精确几何中心坐标 (x_i, y_i)。结合已知的宽高，可唯一确定其四个顶点的绝对位置。
- **优化目标 (Objectives)**:
  - **物理合法性 (Hard Constraint)**: 实现 **无重叠 (Non-overlapping)** 布局。模型需学习处理不同尺寸矩形之间的空间排斥关系，确保所有 Macro 互不干涉。
  - **互连性能 (Performance)**: 最小化 **半周长线长 (HPWL - Half-Perimeter Wire Length)**。通过缩短具有高连接权重的 Macro 之间的几何距离，降低信号延迟和潜在的布线拥塞。

- **核心约束: 高对齐度 (High Alignment)**: 这是本项目的关键差异化目标。布局结果应尽可能规整，Macro 边缘应沿共享的水平或垂直通道线对齐（Snap to shared edges or channels）。这种高度对齐的结构能显著减少布局中的"锯齿"形状，从而简化电源网格设计并提高布线资源的利用率。

3. 数据结构：图表示 (Graph Representation)

模型不处理稀疏的图像矩阵，而是处理紧凑且富含语义的图数据 $G = (V, E)$。

- **节点 (Nodes, V)**: 代表 Floorplan 中的 Macro 实体。
  - **节点特征 (Node Features)**: $F_i = [x_i, y_i, w_i, h_i, \text{type}, \text{connectivity\_density}]$。
    - 其中 (x, y) 是随扩散时间步 t 变化的动态变量，是模型预测的目标。
    - (w, h) 是静态几何特征。
    - type 可以是 One-hot 编码，区分 RAM、IP 核或逻辑模块。
    - $\text{connectivity\_density}$ 聚合了该节点相连的所有 Net 的权重，提示模型该节点的重要性。
- **边 (Edges, E)**: 代表 **Channel Graph** 的拓扑关系，定义了哪些 Macro 在空间上是"相邻"的。
  - 构图方法：边可以基于 **Delaunay 三角剖分 (Delaunay Triangulation)** 构建，捕捉空间邻近性；或基于 **相对位置图 (Relative Position Graph)**，编码"左/右/上/下"的空间约束。
  - 边特征 (Edge Features): 包含通道宽度估计、连接线数 (Wire density/Cut size) 等，用于辅助模型判断两个 Macro 是否应该紧密靠拢。

4. 模型架构：Graph U-Net (The Architecture)

采用 Encoder-Decoder 结构的图神经网络，模仿经典的 U-Net 结构，但算子替换为图算子，旨在从不同尺度的图结构中提取布局特征。

4.1. 整体架构

- **输入层**: 接收带噪的图 G_t，其中节点的坐标特征 x_t, y_t 包含高斯噪声。
- **Encoder (下采样路径)**: 逐层提取全局拓扑特征，减少图的规模。
  - **Graph Convolution**: 使用 **GAT (Graph Attention Network)** 或 GCN。GAT 尤为适合，因为它可以根据连接权重动态调整邻居节点对中心节点的影响力（Attention Weights），模拟 Netlist 中不同连线的重要性。
  - **Graph Pooling (gPool)**: 关键操作。采用 **Top-K Pooling** 策略。
    - **机制**: 利用一个可学习的投影向量 $\mathbf{p}$，计算每个节点特征的投影分数 $score = X\mathbf{p}/\|\mathbf{p}\|$。

    - **操作**: 根据分数保留重要性最高的 $k\%$ 节点，丢弃其余节点，并根据保留节点的连接关系重新构建邻接矩阵。
    - **目的**: 这模拟了图像中的 Downsampling，使模型能够忽略局部微小抖动，转而感知“Macro 簇 (Clusters)”的宏观布局和全局流向。
- **Bottleneck**: 网络的最深层 GNN，处理规模最小但语义最丰富的图，负责决策高度抽象的布局语义（如：哪些簇应该放在芯片中心，哪些在边缘）。
- **Decoder (上采样路径)**: 逐层恢复图的规模，恢复细节并预测位置修正。
    - **Graph Unpooling (gUnpool)**: 关键操作。利用 Encoder 阶段记录的节点索引，将之前被丢弃的节点恢复回来（特征初始化为 0 或通过邻居插值恢复），使图恢复到上一层的规模。
    - **Skip Connections**: 模仿 U-Net 的经典设计，将 Encoder 对应层级之前的节点特征直接 Concat 到 Decoder 对应层。这对于位置预测任务至关重要，因为它保留了局部细节信息，防止在深层特征提取中丢失精确的空间坐标感。
- **输出层**: MLP Head，基于最终恢复的全尺寸图，输出预测的噪声 ϵ 或直接输出去噪后的坐标 (x_0, y_0)。

5. 生成机制：Diffusion Process (The Mechanism)

将 Macro 的坐标 (x, y) 视为连续变量进行扩散，模拟从有序布局到随机热噪声的过程，并学习其逆过程。

5.1. 前向过程 (Forward SDE)

- 保持图拓扑 E 和 Macro 尺寸 (w, h) 不变（因为它们是设计约束）。
- 仅对坐标 $X = \{(x_1, y_1), ..., (x_N, y_N)\}$ 添加高斯噪声：
  $q(X_t|X_{t-1}) = \mathcal{N}(X_t; \sqrt{1-\beta_t}X_{t-1}, \beta_t I)$
- 当 $t \to T$ 时，Macro 的位置信息完全丢失，布局变为完全随机的散点分布，覆盖整个画布空间。

5.2. 反向过程 (Reverse Denoising)

- 利用 Graph U-Net 进行去噪预测：$\mu_\theta(X_t, t, G_{structure})$。
- **物理直觉**: 模型学习的是一个“力场”。在当前乱序的 Channel Graph 状态下，模型预测每个 Macro 应该受到什么样的“力”（移动向量），才能使其从无序状态逐步回归到满足网表连接逻辑和物理约束的合理布局位置。

6. 核心策略：Alignment Guidance (The Solution)

为了解决用户特别强调的 "Alignment" 问题，不能仅依赖模型的隐式学习（因训练集可能对齐度不够完美），必须在推理采样阶段引入显式引导。

采用 **Energy-Guided Sampling (或 Classifier Guidance)**： 在反向生成的每一步 t，计算对齐能量函数的梯度，并将其作为额外的“力”叠加到模型的预测上，修正采样方向。

- **对齐损失函数 (Alignment Energy Function)**:

$$E_{align}(X) = \lambda_1 \underbrace{\sum_i \min_k |x_i - grid_k|}_{\text{Snap-to-Grid}} + \lambda_2 \underbrace{\sum_{(i,j)\in E} \mathbb{I}(aligned) \cdot dist(m_i, m_j)}_{\text{Channel Alignment}}$$

  - **第一项 (Snap-to-Grid)**: 惩罚偏离对齐网格线的坐标。$grid_k$ 是一组预定义的对齐参考线（如基于 Pitch 的网格）。该项迫使 Macro 的中心或边缘“吸附”到最近的网格线上。
  - **第二项 (Channel Alignment)**: 鼓励 Channel Graph 中相邻的 Macro 共享中心线或边缘线。$\mathbb{I}(aligned)$ 是一个指示函数，当两个 Macro 在拓扑上相邻且位置接近时激活，促使它们在水平或垂直方向上精确对齐，形成整齐的行列结构。

- **采样修正 (Sampling Refinement)**:

$$\hat{X}_{t-1} = \mu_\theta(X_t) - \alpha \nabla_{X_t} E_{align}(X_t)$$

  (即：在 Graph U-Net 预测的移动方向上，人为地增加一个基于梯度的拉力。$\nabla_{X_t} E_{align}(X_t)$ 指示了如何微调坐标可以最快降低不对齐的能量，从而将 Macro 强行拉向对齐状态)。

7. 技术栈建议 (Recommended Stack)

- **Framework**: **PyTorch** - 深度学习的标准框架，提供强大的自动求导机制，这对于计算 Guidance 的梯度至关重要。
- **Graph Library**: **PyTorch Geometric (PyG)** - 这是实现 Graph U-Net 最成熟的库。
  - 关键模块: `GATConv` (用于卷积), `TopKPooling` (用于下采样), `Batch` (用于处理图数据的 Batching)。
- **Diffusion Library**: **Diffusers** (可作为调度器和 Pipeline 的参考) 或手写简单的 **DDPM/DDIM** 循环。手写循环更容易集成自定义的 Alignment Guidance 逻辑。
- **Data Handling**: 编写 Parser 解析 **DEF/LEF** 工业标准文件，提取几何信息和网表，构建 PyG 所需的 `Data` 对象。

8. 给 Agent 的执行指令 (Instructions for Agents)

1. **数据预处理 Agent**: 请编写代码将 Floorplan (Macro list + Netlist) 转换为 PyG 的 `Data` 对象。你需要处理特征归一化（将坐标映射到 [-1, 1]），并构建 `edge_index` 以表示 Channel Graph。请确保 `x` 包含所有静态和动态特征。
2. **模型构建 Agent**: 请基于 PyTorch Geometric 实现 `GraphUNet` 类。必须包含 `TopKPooling` 用于下采样，并正确实现 Skip Connections（在 Unpooling 后将 Encoder 的特征 Concat 或

Add 过来)。输入输出维度应适配坐标预测任务（输出维度为 2 或 4，视预测中心点还是预测边界而定)。

3. **训练逻辑 Agent**: 请实现标准的 DDPM 训练循环。Loss 函数建议使用预测坐标与真实坐标的 **MSE**。如果收敛困难，可尝试预测噪声 ϵ 而非直接预测 x_0。
4. **推理逻辑 Agent**: 请实现带有 `guidance_scale` 的采样循环（Sampling Loop)。你需要定义一个可微的 `alignment_loss` 函数，并在 `p_sample` (反向去噪) 步骤中，利用 `torch.autograd.grad` 计算该 Loss 对当前输入坐标 X_t 的梯度，并利用该梯度更新 X_{t-1}，实现对齐引导。