MS Simulation

Phase 1: 数据构建与表示 (Data Construction & Representation)

1. **输入端 (Source Input)**
   - **序列处理**：将肽段氨基酸序列 Tokenize 为整数索引。
   - **元数据融合**：将电荷（Charge）和碰撞能量（NCE）通过 Embedding 映射后，作为特殊的 Token 拼接到序列头部。
   - **填充 (Padding)**：将所有输入序列 Pad 到统一长度 L_{max}。
2. **目标端 (Target Output)**
   - **集合化**：提取实验谱图中的前 K 个强峰（Top-K Peaks）。
   - **归一化 (Normalization)**：
     - **m/z**: 将 $0 \sim 2000$ Da 的值除以 2000，映射到 $[0, 1]$。
     - **Intensity**: 将绝对强度除以该图谱的最大强度（Max Intensity），映射到 $[0, 1]$。
   - **目标集合构建**：对于每张图谱，构建一个无序集合 $T = \{(m_j, i_j)\}_{j=1}^{K}$。
   - **目标填充**：由于模型固定输出 N 个预测（例如 $N = 100$），若真实峰数量 $K < N$，则用 $N - K$ 个“空目标”（$\varnothing$）进行填充。这些空目标在计算损失时将被特殊标记。

Phase 2: 模型架构设计 (Model Architecture)

1. **输入序列构建 (Input Sequence Construction)**
   - 模型接收一个拼接序列，逻辑长度为 $L_{max} + N$。
   - **Part A (Peptide)**: 来源于氨基酸 Embedding + 序列位置编码（Peptide Positional Encoding）。这部分代表“上下文”。
   - **Part B (Queries)**: N 个**可学习的向量 (Learnable Embeddings)**。
2. **编码器主干 (The Encoder Backbone)**
   - 采用标准的 BERT-style Transformer Encoder 层（多层 Self-Attention + FFN）。
   - **交互机制**：Self-Attention 允许 Queries 关注 Peptide Tokens（提取断裂信息），同时也允许 Queries 关注彼此。
   - **Masking**：只需 Mask 掉 Peptide 部分的 Padding Token，其余部分（Peptide $\leftrightarrow$ Queries）全互联。
3. **输出切片与投影 (Slicing & Projection)**
   - 经过 Encoder 后，输出张量维度为 $[Batch, L_{max} + N, Hidden]$。
   - **切片操作**：直接丢弃前 L_{max} 个向量，只保留后 N 个对应 Queries 的向量。
   - **多头预测 (Prediction Heads)**：这 N 个向量并行通过三个独立的线性投影层（Linear Projectors）：
     1. **Position Head**: `Linear -> Sigmoid`，输出 $\hat{m} \in [0, 1]$。
     2. **Intensity Head**: `Linear -> Sigmoid`，输出 $\hat{i} \in [0, 1]$。
     3. **Confidence Head**: `Linear -> Sigmoid`，输出 $\hat{p} \in [0, 1]$（代表该预测是否为真峰）。

Phase 3: 匹配与损失计算 (Matching & Loss)

1. **构建成本矩阵 (Cost Matrix Construction)**
   - 对于一个 Batch 中的每对样本，计算 N 个预测峰与 K_{real} 个真实峰之间的两两成本 C_{ij}。
   - **成本公式 (参考 LIPNovo)**：

$$C_{ij} = \lambda_{coord} \cdot |\hat{m}_i - m_j| + \lambda_{int} \cdot |\hat{i}_i - i_j| - \hat{p}_i$$

2. **执行匹配 (Bipartite Matching)**
   - 使用匈牙利算法（如 `scipy.optimize.linear_sum_assignment`）基于 C_{ij} 找到最优索引对。剩下的 $N - K_{real}$ 个预测者被分配给“背景/空目标”。
3. **计算集合损失 (Set Prediction Loss)**
   - **前景损失 (Matched Pairs Loss)**：
     - 对于匹配成功的 Query，计算 m/z 的 **L1 Loss** 和 Intensity 的 **L1 Loss**。
     - 计算 Confidence 的 **Binary Cross Entropy (Target=1)**。

- **背景损失 (Unmatched / Background Loss)**：
  - 对于未匹配的 Query（即匹配到空目标的 Query），**只计算 Confidence 的 Binary Cross Entropy (Target=0)**。

Phase 4: 推理与生成 (Inference & Generation)

1. **前向传播**：输入多肽序列，模型一次性输出 N 个三元组 $(\hat{m}, \hat{i}, \hat{p})$。
2. **反归一化**：将 $\hat{m}$ 乘回 2000 得到真实 m/z，将 $\hat{i}$ 乘回最大强度得到真实强度。
3. **阈值过滤 (Thresholding)**：
   - 设定置信度阈值 τ（例如 0.5）。
   - 保留 $\hat{p} > \tau$ 的预测点。
   - 丢弃 $\hat{p} \leq \tau$ 的预测点（视为噪声或无信号）。
4. **最终输出**：剩下的点集即为预测的质谱图。