1. 方案1: 场景门控

核心思路
让网络自己学习什么时候该强调场景的影响，就是不让人类预先定义场景类别，而是让神经网络自己从数据中学习，哪些 HOI应该更多地受场景影响。

核心变量

变量名	含义
cls_feat	CLIP 的 class_embedding，包含整张图的全局信息
gate_weights	新增的可学习参数，输出 600 维向量，每个值代表对应 HOI 动作受场景影响的程度（0~1 之间）

实现方式
第一步：特征提取
第二步：新增场景门控模块，这个网络自动学习什么时候强调
新增一个小型神经网络（MLP）， 输入：cls_feat（512维），输出：600 维的权重向量，每个值在 0~1 之间
第三步：融合到分数计算
第四步：训练：端到端训练，不需要额外的人工标注。

2. 方案2: 显式场景分类

核心思路
先告诉网络这是什么场景，再根据场景调整预测，即先让模型显式地预测图片属于哪个场景，然后根据场景来调整 HOI 的预测分数。

核心变量

变量名	含义
cls_feat	CLIP 的全局特征
scene_probs	预测的场景概率分布（10 维，归一化后相加为 1）
scene_hoi_prior	新增的可学习参数，一个 [10 × 600] 的矩阵，第 i 行第 j 列表示"在场景 i 下，HOI j 出现的先验概率"

实现方式

第一步：提取 class_embedding
第二步：新增场景分类模块
这个分类器可以随机初始化，让网络自己学习（无监督）或者用预训练的场景分类模型初始化（有监督，效果更好）
第三步：建立场景-HOI 先验矩阵
第四步：计算场景加权的 HOI 先验

3. 方案3: 多层次特征融合
   核心思路
   在文本匹配、视觉特征、最终分数三个层次都加入场景信息，让场景信息在多个地方发挥作用。
   核心变量
   这个方案需要处理三个层次，每个层次需要不同的输入和输出：

层次	**输入**	**输出**	**作用**
文本层	cls_feat + text_features	场景增强后的文本向量	让文本特征根据场景调整
视觉层	cls_feat	HO tokens 的增强因子	根据场景调整视觉特征
分数层	cls_feat	最终分数的调节因子	直接调整最终输出