

# Hotr: End-to-end human-object interaction detection with transformers

## 摘要

近年来随着图像数据喷涌式地增加，从图片等视觉场景中进行视觉理解引起广泛关注。不同于简单的视觉分类和分割任务，人物交互检测是识别图像中一组交互结果的一项任务，它涉及到人与物之间的定位，同时还涉及到人物对之间的关系，这些关系既有简单的关系，也有复杂的语义关系，对于提升视觉场景下的理解有着重大意义。端到端模型作为不需要后处理的单阶段人物交互检测模型，在推理速度上有大大的提升。但现有的端到端人物交互检测模型大多使用到匈牙利算法，而匈牙利匹配的不稳定性使得模型的性能和收敛速度都受到影响。为解决匈牙利匹配不稳定导致的模型性能下降和收敛速度慢的问题，本文基于 HOTR 模型提出了 POS-HOTR 模型，该模型在交互解码器上引入实例解码器的边界框信息作为位置信息，使得解码器的每一个查询能够更好地关注图片的不同区域，输出比较稳定的人物交互检测三元组，从而提高匈牙利匹配的稳定性，加快模型的收敛速度，提高模型性能。所提出的方法在经典的人物交互检测数据集 VCOCO 上也取得了比较好的性能，相较于原先的 HOTR 模型，改进之后的 POS-HOTR 模型在 VCOCO 数据集的场景 1 中提升了 5.3% 的 mAP，而在场景 2 中提升了 1.1% 的 mAP。

**关键词：**人物交互检测；端到端训练；单阶段模型

## 1 引言

计算机视觉中的人物交互检测指的是一种技术，即在图像或视频中定位并识别出人体、物体，并准确地检测和描述他们之间的交互关系。人物交互检测可以帮助计算机更好地理解图像和视频中的人物交互场景，从视觉场景中获取更多信息。人物交互检测区别于目标检测在于，除了进行人物的检测和定位，还要进行人物之间的关系检测，同时还有这个关系应该对应的人与物之间的检测，这个是比较困难的问题。图 1 所示是常见的人物交互检测的标注方式，通常是标注人与物的位置，同时物还有类别，然后对这些人物对，判断哪些人物对之间是有关系的，关系是什么。

人物交互检测算法针对时空人体动作进行更加精细化的理解，是一种更高层次的视觉场景内容理解技术。近些年，随着深度学习的发展以及大规模数据库的提出，人物交互检测算法逐渐受到了研究者的关注。该任务有非常广泛的应用场景，其研究成果可以广泛应用于视频监控、智能车舱、人机交互和智能机器人等应用情景，也可以支撑起更高级的视觉内容理解任务，如视觉稳定、图像描述等。人物交互检测算法通过预测 HOI (Human-object interaction)

三元组来预测人物对的关系，不同于传统的空间位置关系预测，人物交互检测算法更加关注广泛的行为关系与联系关系，可以更加精细地刻画图像信息，更加有效地进行视觉内容理解。

## 2 相关工作

在深度学习模型上人物交互检测的表达为：对于输入的图片  $I$ ，输出一个集合，这个集合记录人-物位置以及它们之间的交互关系， $\{HOI\text{三元组}\}_{i=1}^K$ ，这个集合就是一个 HOI 三元组的集合，HOI 三元组的表示方式为  $\langle box_h, box_o, act_c \rangle$ ，分别是人物之间的定位，和彼此之间的关系，需要注意的是，这种 HOI 三元组的表示方式其实是淡化了物体的类别，而人只有一个类别不需要判断，这是在人物交互检测的目标检测任务上有别于单纯目标检测任务的地方。人物交互检测中，根据进行人物对匹配的方式不同可以分为双阶段人物交互检测算法、单阶段人物交互检测算法、端到端人物交互检测算法。

### 2.1 双阶段人物交互检测模型

双阶段人物交互检测算法，顾名思义需要分两个过程进行，第一个过程是进行目标检测，即通过精细调整的目标检测器来获取到人和物的边界框和类别标签。对于第一个阶段预测出来的高置信度的人和物体，通过两两组合或者构造成图的方式作为候选集，接下来就进入了第二阶段，使用交互类别分类器进行人物对的类别分类。由于第一阶段的任务主要是目标检测，因此当前更多的研究是倾向于第二阶段的研究，即对交互类别分类器的研究。

通常，在上述多流体交互分类器中有三个流：人流、物体流和成对流。人流和物体流通常分别对人和物体框进行视觉特征编码 [4]。在 FCMNet [12] 中，由于物体的详细视觉外观通常对于交互类别不是至关重要的，因此将物体视觉特征替换为单词嵌入。除视觉特征外，Bansal 等人 [1] 还在人流中引入了单词嵌入进行特征增强；PDNet [20] 引入了单词嵌入到所有流中，以获得语言先验引导的通道注意力和特征增强。关于成对流，进行了大量研究。该流通常编码人和物体之间的关系。首先在 iCAN [4] 中提出了两通道二进制图像表示来编码空间关系，但在 FCMNet [12] 中，从人体解析中提出了一种细粒度版本来放大关键提示。除空间关系外，DRG [3]，CHG [17]，RPNN [21] 中的图神经网络提出了显式建模人与对象之间交互的方法，这确实提高了模型的表示能力。辅助模型可以轻松引入两阶段管道，以帮助改进 HOI，例如人体姿态特征，人体部分 [9]，语言模型 [14] 和图模型 [19] 等。有趣的是，Bansal 等人 [1] 和 Hou 等人 [6] 引入了特征级增强，证明对于 HOI 非常有效。然而，由于顺序和分离的两阶段架构，这些方法复杂度高且效率低。

如图1所示，双阶段人物交互检测算法是实例驱动的交互检测算法，需要预先检测出  $M$  个人和  $N$  个物体，然后对这些人物进行组合，从而得到  $M \times N$  个人物对，接下来对这个人物对的集合进行预测，因此计算量比较大，存在一些明显的问题。例如进行穷极组合的人物对，其中真正拥有关系的实例对是比较少的，这导致正负样本不平衡，容易导致网络过拟合，输出高置信度的“无交互”，抑制了正样本的分类结果；其次，在这个过程中对穷举的人物对进行检测，实际上是浪费了计算资源，因为大多数人物对并没有关系。当然两个阶段分开进行检测的模型，虽说可以在第一个阶段将目标检测任务做到很好，然后再基于此进行第二阶段的交互检测检测，但事实上两个过程是可以相互纠正的，第二阶段缺少的人物对上下文信息，可能会导致分类出现一定的歧义性。

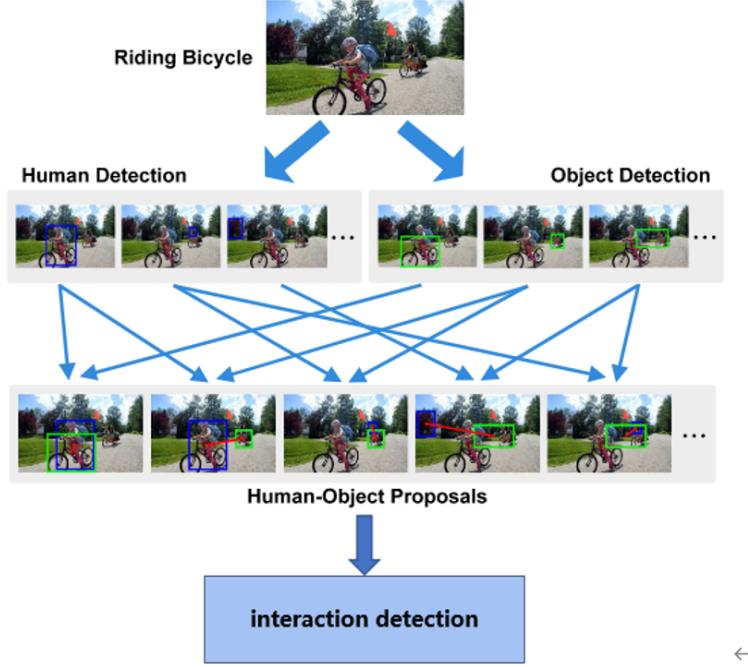


图 1. 双阶段人物交互检测常见范式

## 2.2 单阶段人物交互检测模型

与双阶段人物交互不同，单阶段人物交互检测算法将人物关系的检测结合在一个网络之中，当然这是最通常的定义，更加严格的定义应该是不使用穷举人物对的方式进行交互检测的人物交互检测算法才被称为人物交互检测。单阶段人物交互检测算法会想方设法去避开穷举人物对，例如 PPDM [10] 和 IPNet [18]，将 HOI 视为一个点检测问题，通过引入交互点的概念，将交互的判断也视为点的判断，在得到交互点了之后，便利用交互点进行人物对的匹配，这个复杂度是线性的，因此相比于人物对的穷举组合，可以节省比较多的计算代价。此外，PPDM [10] 是在一个统一的 CenterNet-based [22] 模型中进行的 HOI 检测。而在 UnionDet [7] 中，HOI 检测被视为一个组合框检测问题，这种基于框的单阶段预测模型，摒弃了两阶段模型中比价耗费时间的交互检测器，在检测人物的同时并行地进行可能发生交互检测关系的边界框，通过这些边界框包含的人物对，完成 HOI 三元组的组合。这一类模型基本基于流行的 RetinaNet [11]，将额外的组合分支添加到传统的物体检测分支旁边以检测组合框。

图2展示了单阶段人物交互检测算法的基本形式，一般是两个并行分支，一个分支进行人与物的目标检测任务，完成人物的定位和分类，另一个分支进行关系的预测，同时预测关系应该指向的人与物，通过这个指向完成每一个关系应该匹配的人与物，从而预测得到最终的 HOI 三元组。

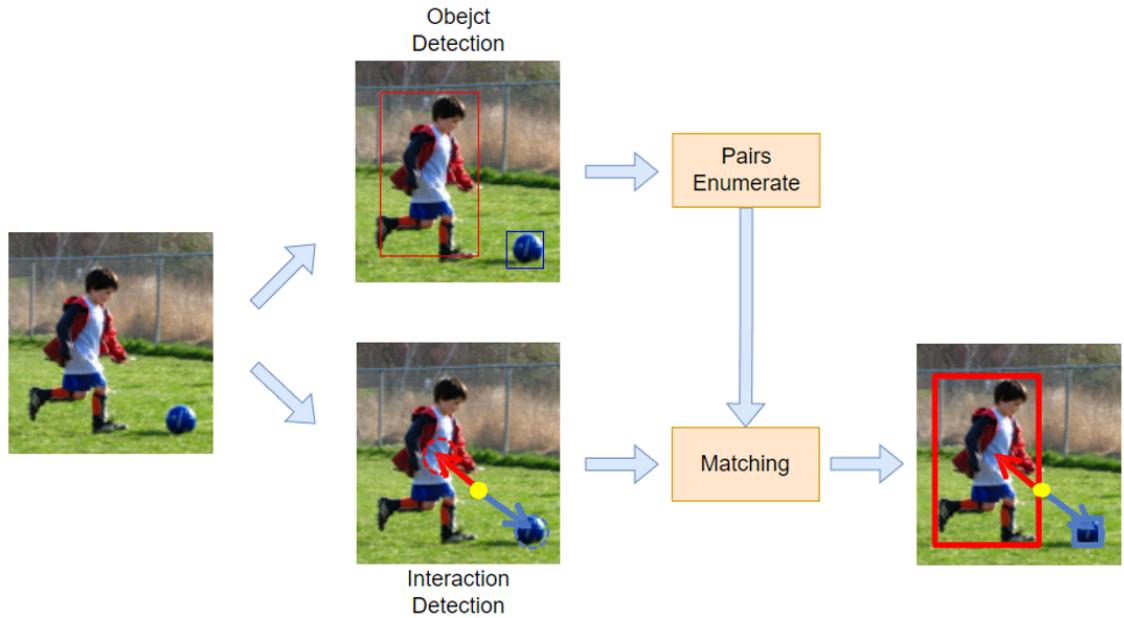


图 2. 双阶段人物交互检测常见范式

与两阶段方法相比，流程变得更简单、更快、更高效，更易于部署到实际应用中，并且精度也有所提升，这是得益于自顶向下的结构，直接定位了关系的点或者框，再进行人物对的匹配，避免了负样本的人物对进行关系检测的影响，进而提高了精度。然而，单阶段方法仍然需要复杂的后处理来组合物体检测结果和交互预测。

### 2.3 端到端人物交互检测模型

端到端模型其实也是单阶段模型人物交互检测模型的一类，不过与传统的单阶段模型不同，端到端方法通常是没有非极大值机制体系结构，即不需要进行麻烦的后处理，这对于推理速度的提升是非常有帮助的，可以得到更快的推理速度。最初 Russell [15] 提出了一个基于 LSTM 编码器-解码器的端到端行人检测方法，这是一种自回归模型，逐个元素预测输出序列，这种利用 RNN 类似结构进行端到端预测的模型，通常受限于 RNN 不能并行运算的缺点，尽管去除了 NMS 体系结构，但是在推理速度上还是不太理想。在目标检测领域中率先出现的端到端预测模型 DETR [2]，通过将 LSTM 替换为 Transformer [16] 来进行改进，将这一类预测问题转变为集合预测问题 [13] [5]，这对于人物交互检测算法也有不少启示，借助最近的具有并行解码的 Transformer，可以并行解码 N 个对象，在人物交互检测领域即并行解码 N 个 HOI 三元组。这些方法都需要使用匈牙利算法来进行真实结果与预测结果的匹配，在这些结构中，匈牙利匹配似乎是迄今为止最好的选择。

但目前端到端人物交互检测算法也依然存在一些问题，由于匈牙利匹配的不稳定性，模型的每一次反向传播都可能产生不同的匹配结果，使得模型训练在收敛速度和性能上可能有所下降。针对于此，本文通过对端到端模型中的 Transformer Decoder 进行优化，具体是为其添加位置信息，达到提高模型收敛稳定性和性能的效果。

### 3 本文方法

#### 3.1 本文方法概述

本篇论文构建的模型为 HOTR(Human-Object interaction Transformer)，这个模型利用的是 Transformer 的架构。给定一张图片的输入，首先会通过卷积神经网络 Backbone 进行特征提取，同时将图片拆分为多个块 (Patch)，接下来将块输入到共享编码器 (Shared Encoder) 进行图片上下文信息的聚合，使得每一个图像块也同时拥有其他图像块的信息。经过共享解码器的特征凝聚之后，数据将会流动到两个并行的解码器，分别为实例解码器 (Instance Decoder) 和交互解码器 (Interaction Decoder)。实例解码器的架构与 DETR [2] 相类似，主要完成的是目标检测任务，即获取到图像中人和物的中心位置和边框信息。交互解码器完成的是对交互类别的预测，预测可能发生什么样的动作，同时为该动作可能发生在哪一个人和物进行预测(在本文中称为人物指针,Human Poitner 和 Object Pointer)。最终汇聚两个并行解码器的信息，通过交互解码器预测出来的人物指针进行人和物的指向，为交互动作分配应该指向的人和物，从而组成 HOI(Human Object Interaction) 三元组，即  $\langle box_h, box_o, act_c \rangle$ 。

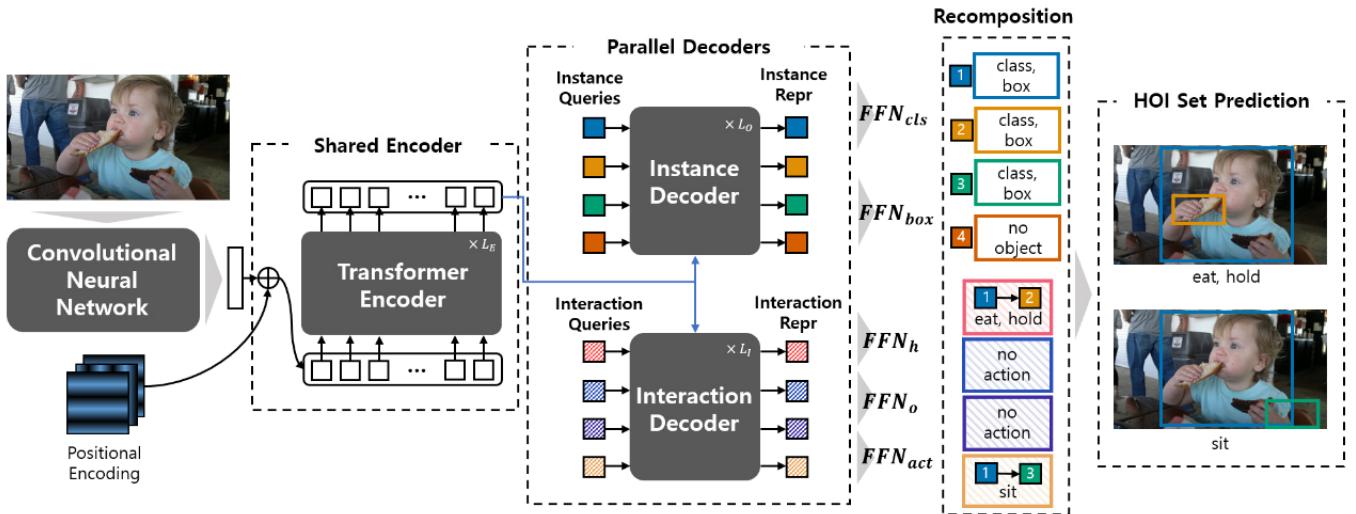


图 3. 模型的总体流程，实例解码器 (Instance Decoder) 和交互解码器 (Interaction Decoder) 并行运行，并共享编码器，而在重组 (Recomposition) 过程，会将目标实例和关系进行匹配，从而得到 HOI(Human Object Interaction) 三元组。

#### 3.2 HOTR 解码器架构

在图3中，将输入图像进行分块以及使用 Transformer 的共享编码器进行特征提取是一个非常经典的流程，详细可以参考 DETR [2] 中的做法，因此这一部分重点讲述的是与之不同的部分，即两个解码器是如何进行解码和重组的。

如图4两个解码器都是首先学习到输入进来的每一个查询的嵌入向量 (Instance Repr 和 Interaction Repr)，而实例解码器则是将得到的嵌入向量输入到两个 MLP(Multilayer Perceptron) 中去，其中一个 MLP 预测类别，即是人还是物，是物体的话为哪一个物体，另一个 MLP 预测空间位置，即边界框 (Box)。交互解码器做的事情与实例解码器有所不同，交互解码器将学习到的嵌入向量输入到三个 MLP 中去，第一个 MLP 进行交互类别的预测，第二个

MLP 预测应该指向的人的嵌入向量值，第三个 MLP 预测应该指向的物的嵌入向量值，然后通过预测嵌入向量值与实例解码器中的嵌入向量进行归一化的余弦相似度近似匹配，找到该交互动作应该指向的人和物体。

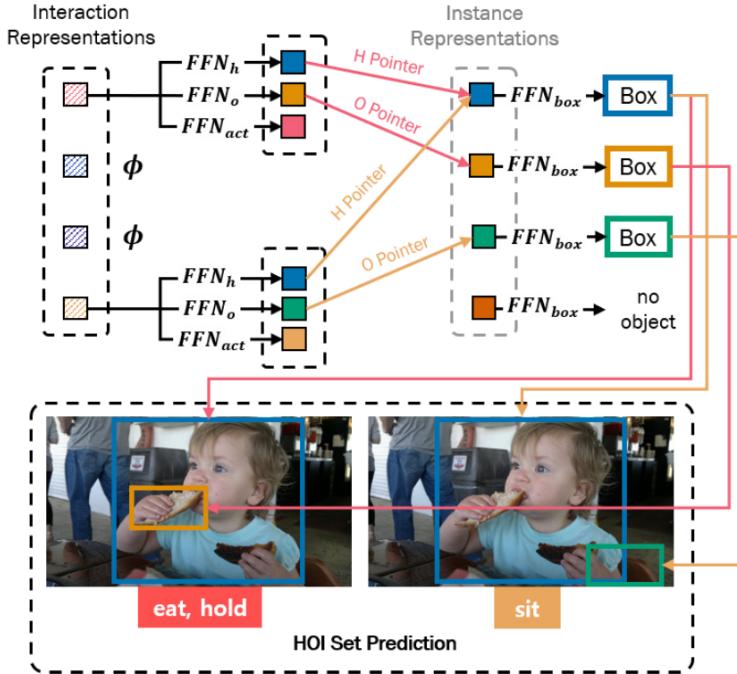


图 4. 这幅图用于阐述人物指针是如何与交互关系进行组合的。这里的人物指针学习的并不是一个简单的指向，而是一个与实例解码器中学习得到的嵌入向量相同维度的嵌入向量，因此匹配的时候是通过两个嵌入向量的相似度进行匹配的，而衡量两个嵌入向量的相似度是利用余弦相似度的方法进行比较的。

### 3.3 损失函数定义

集合预测模型在进行损失函数计算之前，需要将一个无序集合预测出来的结果与真实的结果进行匹配，才能得到预测结果与真实结果一一对应的关系，因此在进行损失函数之前需要先进行二分图匹配，而在 HOTR [8] 中使用的是匈牙利算法进行二分图匹配，使得无序集合中的每一个预测结果都能有唯一对应的真实标签。

在完成了匈牙利匹配之后，模型就得到了真实结果与预测结果之间的配对，此时才可以使用损失函数进行计算，考虑到 HOI 三元组需要有位置预测结果的约束和动作类别之间的约束，因此损失函数的设计如公式 (1) 和公式 (2) 所示：

$$\mathcal{L}_H = \sum_{i=1}^K [\mathcal{L}_{loc}(c_i^h, c_i^o, z_{\sigma(i)}) + \mathcal{L}_{act}(a_i, \hat{a}_{\sigma(i)})]. \quad (1)$$

其中位置损失函数为

$$\begin{aligned} \mathcal{L}_{loc} = & -\log \frac{\exp \left( \text{sim} \left( \text{FFN}_h(z_{\sigma(i)}), \mu_{c_i^h} \right) / \tau \right)}{\sum_{k=1}^N \exp \left( \text{sim} \left( \text{FFN}_h(z_{\sigma(i)}), \mu_k \right) / \tau \right)} \\ & -\log \frac{\exp \left( \text{sim} \left( \text{FFN}_o(z_{\sigma(i)}), \mu_{c_i^o} / \tau \right) \right)}{\sum_{k=1}^N \exp \left( \text{sim} \left( \text{FFN}_o(z_{\sigma(i)}), \mu_k \right) / \tau \right)} \end{aligned} \quad (2)$$

## 4 复现细节

### 4.1 与已有开源代码对比

此部分为必填内容。如果没有参考任何相关源代码，请在此明确申明。如果复现过程中引用参考了任何其他人发布的代码，请列出所有引用代码并详细描述使用情况。同时应在此部分突出你自己的工作，包括创新增量、显著改进或者新功能等，应该有足够差异和优势来证明你的工作量与技术贡献。

HOTR [8] 作者已经完成了代码的开源，代码地址为<https://github.com/kakaobrain/hotr>。因此在实验的过程中不只是将作者的代码环境搭配完成并且完成训练如此简单，在复现中注意到作者的模型依然是存在一定的不足之处，因此会提出一些方法对作者的模型进行改进并且通过实验验证改进的效果；同时作者的开源代码中并没有对人物交互检测任务进行可视化，我也在代码中实现了这一部分的功能，使得人物交互检测这一个任务可以更加直观地看到效果。

### 4.2 实验环境搭建

本次实验环境使用的是趋动云平台的云算力环境，具体是带有 4 个 RTX3090GPU 的服务器。深度学习框架使用的是 pytorch 框架。

### 4.3 创新点

HOTR 模型利用的是匈牙利匹配算法进行预测结果与真实结果的匹配，而匈牙利匹配的不稳定性使得模型的性能有所下降，并且收敛速度也比较缓慢。针对于 HOTR 模型匈牙利匹配的不稳定性，本文设计了一种新的解码器，即 POS 解码器，通过引入人和物的边界框信息，同时为交互动作实体化，为解码器提供稳定的位置信息，使得解码器的查询可以关注图像上下文信息的同时也关注到图像固有的位置信息，进而匈牙利匹配的稳定性，防止无效的匈牙利匹配浪费性能。

POS 解码器的结构如图5所示，相比于原先的交互解码器，POS 解码器并不是在最终阶段才进行人物指针的解码的，而是在过程中就已经将这些给解码。对人物指针进行解码可以获取到人和物的边界框信息、位置信息，通过位置编码等操作将这些位置信息添加到交叉注意力阶段的查询中去，完成了额外信息的使用，这些额外信息的使用可以有效地提高匈牙利匹配的稳定性，使得每一个查询可以关注于图像的特定区域，不至于反复摆动导致匹配结果变化不断。

本文设计的 POS-HOTR 细节比较多，包括了位置信息获取、正余弦位置编码、自注意力查询改进、交叉注意力查询改进等部分，由于篇幅问题就不在这里做具体展开，可以从图5中获取到这些细节，或者从代码中去查看这些细节的实现。图5 中紫色的部分即为设计与创新的部分。

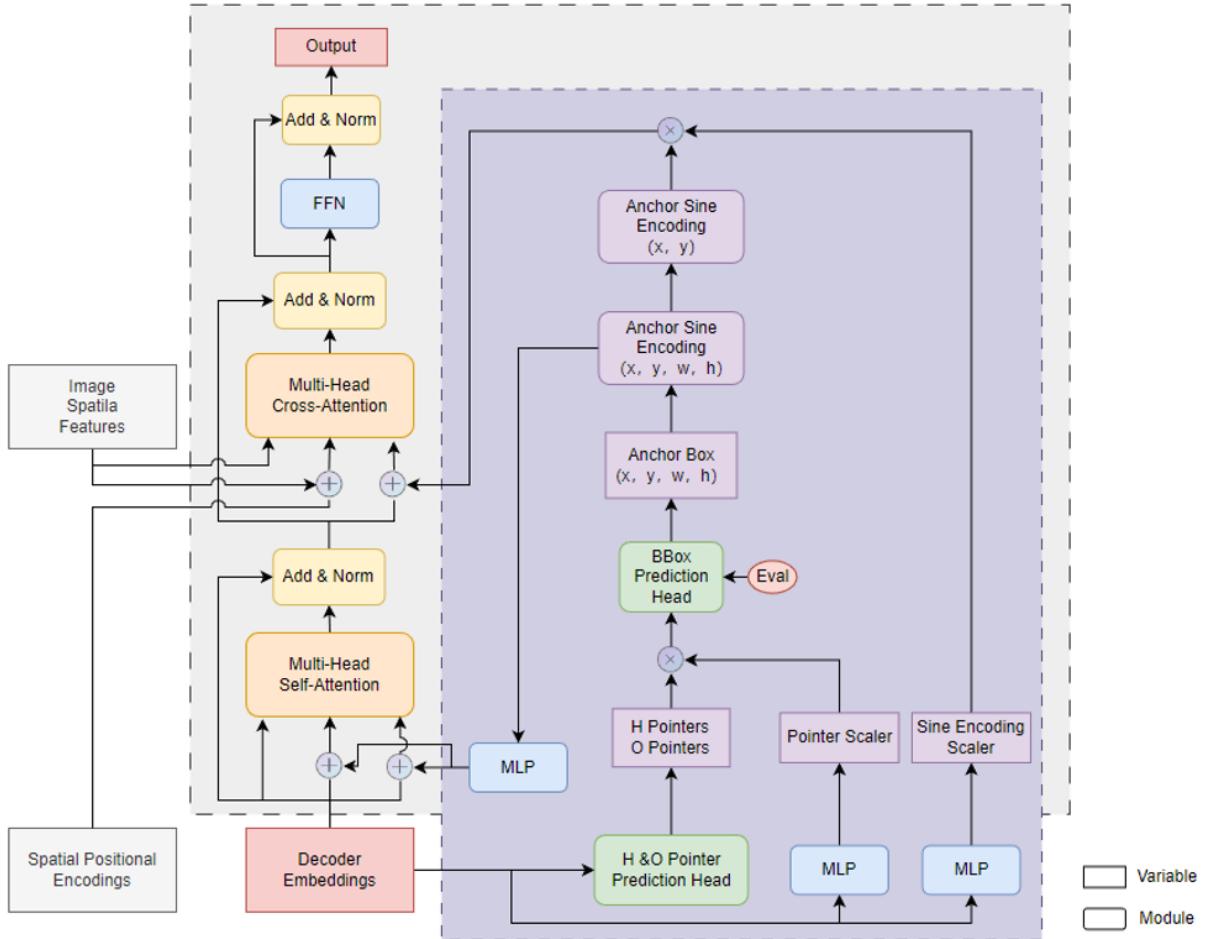


图 5. POS 解码器示意图。POS 解码器在首先会通过初始的嵌入向量获取到对应的人物指针 (H&O Pointer)，接下来通过人物指针解码出来人和物体对应的边界框，同时将人和物的边界框信息进行融合，最后将融合信息进行正余弦位置编码得到位置信息的嵌入向量，将该嵌入向量添加到交互解码器的第一个输出中。

## 5 实验结果分析

本部分对实验所得结果进行分析，详细对实验内容进行说明，实验结果进行描述并分析。

### 5.1 数据集

VCOCO (Visual Compositional COCO) [35] 是一个针对人物交互检测任务的数据集，它是在 COCO (Common Objects in Context) 数据集的基础上构建的。VCOCO 数据集包含 10,346 张图像，其中 5,974 张用作训练集，2,783 张用作验证集，1,589 张用作测试集。每张图像中都包含一个或多个个人物以及他们之间的交互。数据集中的每个人物与其它人物之间的交互都被分类为 49 个不同的动作类别，这些类别包括“打电话”，“拥抱”，“接电话”，“摔倒”等等。此外，VCOCO 数据集还提供了每个人物所处的位置和姿势信息，这些信息可以帮助研究人员更好地理解交互过程。

VCOCO 数据集中的每个图像都被注释了一个人物的 bbox 和其它人物之间的交互关系。每个交互都由两个人物的 ID 和一个动作类别组成。此外，VCOCO 数据集还提供了每个人物

的关键点信息，用于提高模型的姿态估计能力。该数据集的研究意义在于，它为人物交互检测任务提供了一个标准测试基准，并且可以帮助研究人员更好地理解人物交互的动作和姿势。VCOCO 数据集已经被广泛应用于人物交互检测、动作识别、视频分析等领域的研究中。

在 VCOCO 数据集中有两个场景，具体如图6和图7所示，分别是场景 1 (Scenario1) 和场景 2 (Scenario2)。Scenario1：在这个场景中，图像中至少有两个人，并且他们正在进行一个特定的动作。VCOCO 数据集中包含了 16 种动作。Scenario2：在这个场景中，只有一个人出现在图像中，但他/她正在和环境中的某些物体进行互动。VCOCO 数据集中包含了 10 种动作。

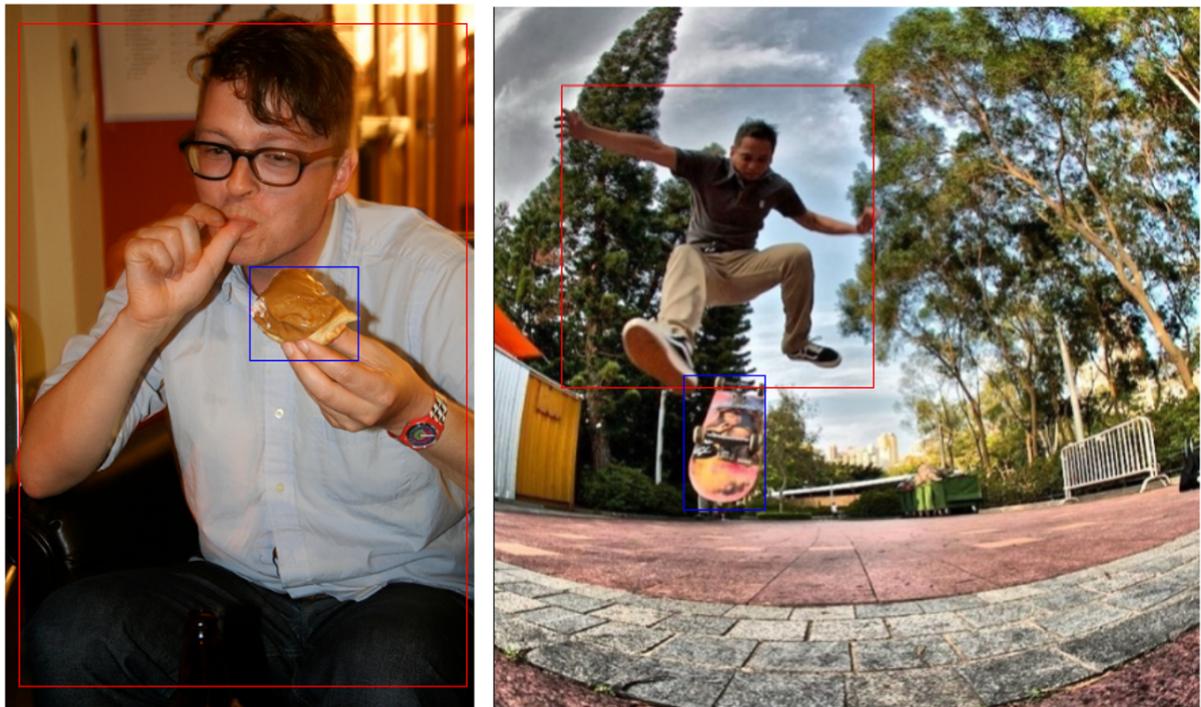


图 6. VCOCO 数据集中场景 1 (Scenario 1) 示例样本



图 7. VCOCO 数据集中场景 2 (Scenario 2) 示例样本

## 5.2 评价指标

在人物交互检测中也经常使用 mAP 进行模型效果的评价，与目标检测存在一定的相似度，但是又有所不同，mAP (mean Average Precision)，它其实是各个类别 AP 的平均值，在人物交互检测中即为各个交互类别 AP 的平均值，AP 值指的就是 PR 曲线下与坐标轴包围的面积。其中 P 是查准率，而 R 是查全率。

对于一个预测出来的 HOI 三元组  $\langle \text{box\_h}, \text{box\_o}, \text{action} \rangle$ ，由于边界框与实际的边界框之间是存在一定的差距的，这个差距的大小一般使用 IOU (intersection of union) 来衡量。IOU 意为交并比，是在目标检测和图像分割领域中经常使用的指标，计算公式如下

$$IOU = \frac{\text{box}_{gt} \cap \text{box}_{pre}}{\text{box}_{gt} \cup \text{box}_{pre}}$$

其中  $\text{box}_{gt}$  代表的是真实的边界框，而  $\text{box}_{pre}$  代表的是预测的边界框。在拥有了 IOU 的概念之后，就可以开始定义人物交互检测中的 TP、FP 和 FN

- TP: True Positive。表示人的边界框和物的边界框大于阈值，一般阈值设置为 0.5，同时交互类别预测也是正确的，才会被判为 TP，真阳性。并且是要满足一个真实 HOI 三元组只被计算过一次，当多个 HOI 三元组与真实 HOI 三元组的 IOU 大于阈值的时候，取 IOU 最大的那一个。
- FP: False Positive。表示人的边界框和物的边界框其中一个小于阈值，或者交互类别预测错误的预测结果数量，又或者同一个 HOI 三元组的冗余预测 HOI 三元组的数量。
- FN: False Negative。表示没有检测到的 HOI 三元组的数量。

查准率 (Precision) 顾名思义，在所有被检测出来的 HOI 三元组中，判断为正阳性的 HOI 三元组的比例，起到的是评判所有检测出来的 HOI 三元组中，有多少是正确的，计算公式为

$$\text{Precision} = \frac{TP}{TP + FP}$$

查全率 (recall)，是在所有的真实 HOI 三元组中，判断为正阳性的 HOI 三元组的比例，起到的是评判在所有真实 HOI 三元组中，有多少是被检查出来的，计算公式为

$$\text{Recall} = \frac{TP}{TP + FN}$$

接下来对于每一个交互类别下预测的 HOI 三元组，根据置信度进行排序，然后一步一步计算得到 Precision 和 Recall，就得到了 PR 曲线，再利用微积分近似计算方法求解面积，即得到 AP，对于各个交互类别的 AP 求均值，得到的即为人物交互检测的 mAP。

## 5.3 实验设置

本文模型的设计有两个目标，一个是提升模型的性能，即提高模型在数据集测试上的 mAP，另一个则是加快模型的收敛速度，表现为相同参数的情况下，模型的 mAP 提升与损失下降比原先的模型要更加优秀。

实验设计上，除了交互解码器有无添加位置信息的差别，其他参数完全一致，backbone 的学习率设置为  $10^{(-5)}$ ，而其他模块的学习率设置为  $10^{(-4)}$ ，batch size 的大小设置为 2，利用趋动云平台使用 4 个 GPU 同时训练。

在测试模型性能的时候，这里是训练了 100 个 epoch 进行比较，主要差别是在 epoch 为 85 的时候会进行学习率的衰减，这个衰减对于性能是有一定的影响的，因此两个模型的性能比较是在 100 个 epoch 进行比较。而在进行模型收敛速度比较的时候，为了节约算力，采用的是进行前 20 个 epoch 收敛速度的比较，因为到了训练后期虽然依然有性能上的差距，但是两者收敛速度比较一致，因此在实验设计上就只进行前 20 个 epoch 的比较。

## 5.4 实验结果

表1展示了在具有场景 1 和场景 2 的 VCOCO 数据集下的实验结果，可以观察到，对交互解码器进行了位置信息的添加之后，在 VCOCO 的两个场景下的结果都有所提升，其中在场景 1 下的提升比较明显，提升了 5.3%，而在场景 2 下也有所提升，提升了 1.1%，效果虽不及场景 1 那么明显，但是也是有所提升。由此可以看到在交互解码器中进行位置信息的添加是可以较好地提高模型的性能的。

表1中的 HOI 三元组推理时间是在单个 GPU 下测量的平均推理时间，从模型参数量和推理时间上来看，POS-HOTR 还是有所劣势的，具体表现在模型参数量增多并且推理时间变长。

表 1. POS-HOTR 与 HOTR 在 VCOCO 数据集上 mAP 比较 (这里的  $AP_{role}^{\#1}$  指的是 VCOCO 数据集场景 1 下的 mAP,  $AP_{role}^{\#2}$  指的是 VCOCO 数据集场景 2 下的 mAP)

Epoch	Model	Queries	Interaction Decoder	Params	Inference Time	$AP_{role}^{\#1}$	$AP_{role}^{\#2}$
100	HOTR [8]	16	Normal Decoder	51.18M	46ms	55.2	64.4
100	POS-HOTR	16	POS-Decoder	51.57M	121ms	<b>60.5</b>	<b>65.6</b>

从表 2 与其他的人物交互检测模型进行比较，可以看到 POS-HOTR 的性能与利用 R101 作为 Backbone 的双阶段人物交互检测模型 UPT[36] 非常相近，这个性能上的提升是比较明显的，也侧面说明在交互解码器中进行位置信息添加的效果是非常有效的。

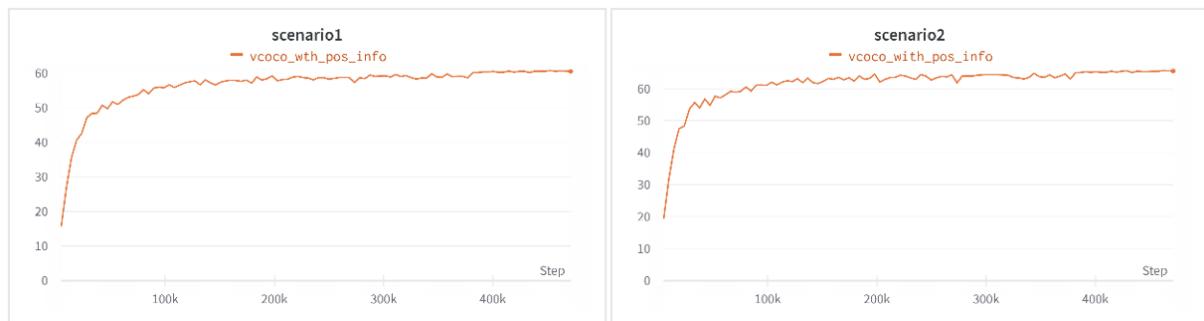


图 8. POS-HOTR 在 VCOCO 数据集场景 1 和场景 2 的 mAP 训练曲线。横坐标为步数，纵坐标是 mAP，单位为百分比

对于交互解码器的改进，除了可以对模型的性能有所提升，还可以加快模型的收敛速度，

表现为在参数不变的情况下，同样的训练步数中，改进之后的模型损失函数的值更低，并且评价指标 mAP 更高，如图9所示，可以看到无论是在 VCOCO 的场景 1 还是 VCOCO 的场景 2 中，添加了位置信息的交互解码器性能都是优于原先的交互解码器。而从损失函数的下降过程也可以看到，添加了位置信息之后的交互解码器损失函数之下降得更快，基本包括在了原先交互解码器曲线的下方。这些数据分析结果都可以支撑进行位置信息添加的交互解码器，在加快模型收敛速度上是有效果的。

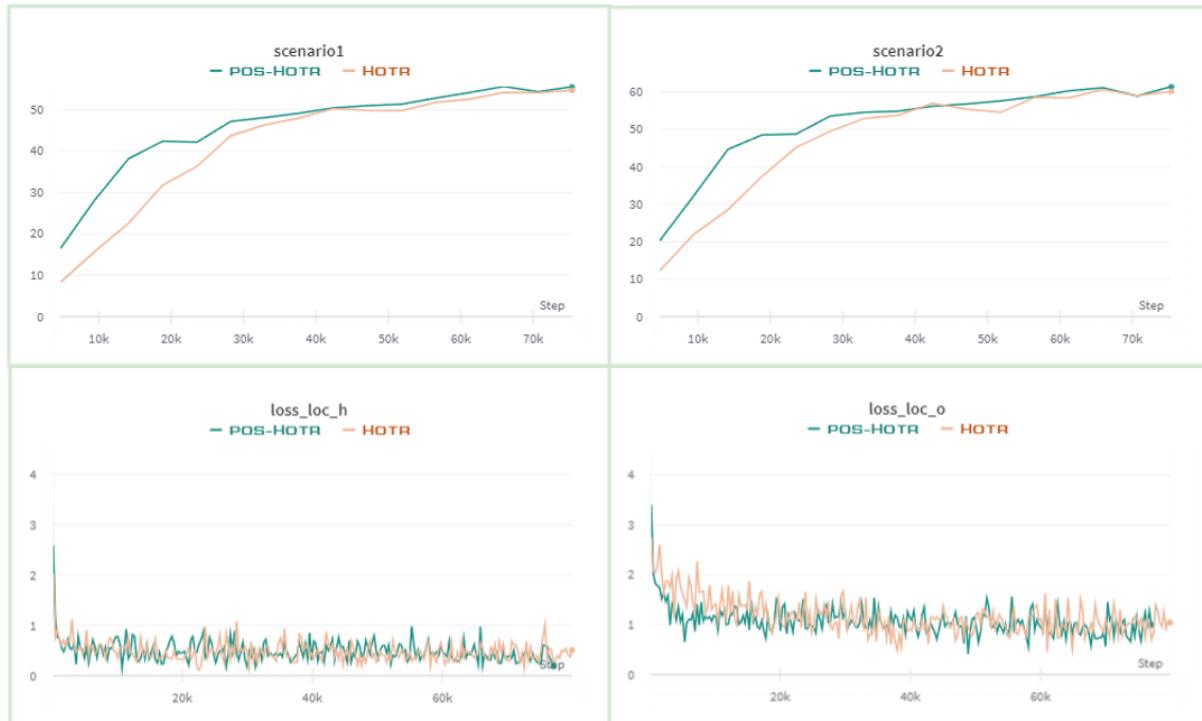


图 9. POS-HOTR、HOTR 在 VCOCO 数据集上场景 1 和场景 2mAP 上升速度以及损失函数值下降速度比较，横坐标为步数，mAP 曲线中纵坐标单位为百分比（左上是 VCOCO 数据集下场景 1mAP 上升曲线、右上是 VCOCO 数据集下场景 2mAP 上升曲线，左下和右下分别是人物指针损失函数值的下降曲线）

## 5.5 可视化分析

在这个部分主要对预测的结果进行可视化分析，图10和图11分别是 POS-HOTR 在 VCOCO 数据集场景 1 和场景 2 的可视化结果，可以看到模型对于多个人物的复杂场景还是可以进行比较不错的人物交互检测的。

图10和图11则是 POS-HOTR 与原先 HOTR 的结果对比，可以看到在关系判断上，POS-HOTR 会更加准确，具体表现在不仅语义更准确了，而且在可以更加准确罗列一个人的各种行为。

POS-HOTR 模型在 VCOCO 数据集上也有一些失败的样例，如图12所示，这些预测失败主要是：无法预测没有出现在训练集上的关系类别、存在语义飘移、缺少图片深度信息。可以看到无论是简单场景还是复杂场景，语义飘移都是比较严重的，这可能是人物关系的长尾分布所导致的。

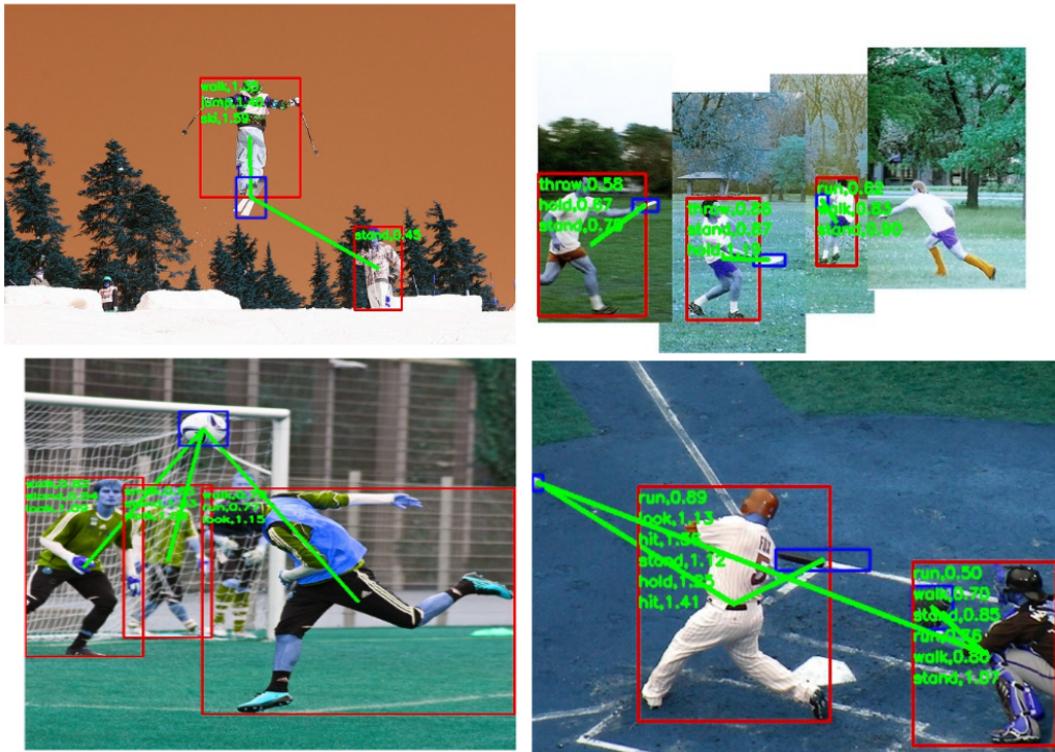


图 10. POS-HOTR 在 VCOCO 数据集场景 1 的部分预测结果



图 11. POS-HOTR 在 VCOCO 数据集场景 2 的部分预测结果

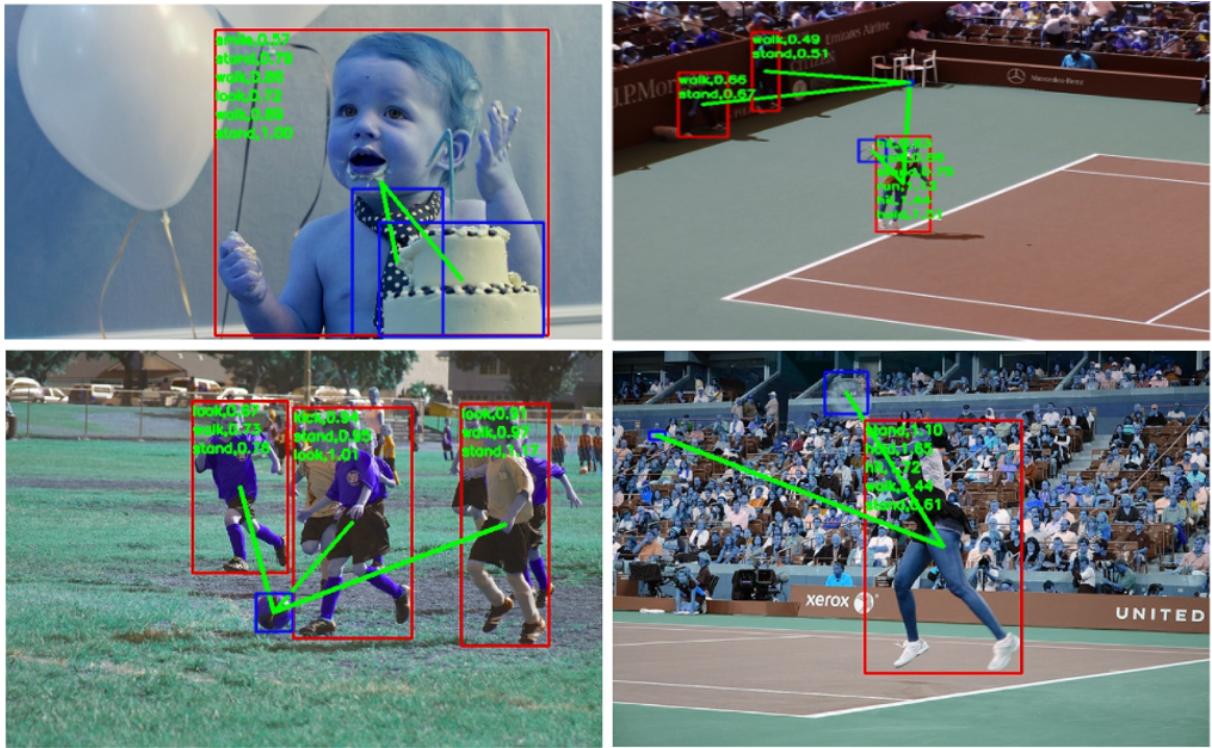


图 12. POS-HOTR 在 VCOCO 数据集上预测效果不好的样例

## 6 总结与展望

端到端人物交互检测算法可以避免后处理，从而在推理速度上大大提升，而目前端到端人物交互检测算法大多没有考虑位置信息的使用，这导致了模型性能的下降与训练中收敛的不稳定。针对于此问题，本文提出了一种在端到端人物交互检测算法上进行位置信息添加的思路，具体是在交互解码器中通过对实例解码器中的人物对进行边界框获取，从而得到位置信息的方法。同时本文还在 HOTR 端到端人物交互检测算法上进行思路实践，通过实验可以知道，添加了位置信息的 POS-HOTR 模型相比于 HOTR 模型，在 VCOCO 数据集的场景 1 中提升了 5.3% 的 mAP，而在场景 2 中提升了 1.1% 的 mAP。该研究表明了位置信息在端到端人物交互检测算法中可以提高模型的性能与收敛速度，在进行模型训练中有着重要意义。本文的不足之处在于设计的模块并非是即插即用的深度学习网络层，因此在实践的时候需要根据网络的结构进行特定的设计与改动，如何设计即插即用的带有位置信息的人物交互检测算法模块也是未来工作的核心。

## 参考文献

- [1] Ankan Bansal, Sai Saketh Rambhatla, Abhinav Shrivastava, and Rama Chellappa. Detecting human-object interactions via functional generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10460–10469, 2020.
- [2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

- [3] Chen Gao, Jiarui Xu, Yuliang Zou, and Jia-Bin Huang. Drg: Dual relation graph for human-object interaction detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 696–712. Springer, 2020.
- [4] Chen Gao, Yuliang Zou, and Jia-Bin Huang. ican: Instance-centric attention network for human-object interaction detection. *arXiv preprint arXiv:1808.10437*, 2018.
- [5] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.
- [6] Zhi Hou, Xiaojiang Peng, Yu Qiao, and Dacheng Tao. Visual compositional learning for human-object interaction detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 584–600. Springer, 2020.
- [7] Bumsoo Kim, Taeho Choi, Jaewoo Kang, and Hyunwoo J Kim. Uniondet: Union-level detector towards real-time human-object interaction detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 498–514. Springer, 2020.
- [8] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 74–83, 2021.
- [9] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Shiyi Wang, Hao-Shu Fang, Ze Ma, Mingyang Chen, and Cewu Lu. Pastanet: Toward human activity knowledge engine. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 382–391, 2020.
- [10] Yue Liao, Si Liu, Fei Wang, Yanjie Chen, Chen Qian, and Jiashi Feng. Ppdm: Parallel point detection and matching for real-time human-object interaction detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 482–490, 2020.
- [11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [12] Yang Liu, Qingchao Chen, and Andrew Zisserman. Amplifying key cues for human-object-interaction detection. In *European Conference on Computer Vision*, pages 248–265. Springer, 2020.
- [13] Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al.

- Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*, pages 3918–3926. PMLR, 2018.
- [14] Julia Peyre, Ivan Laptev, Cordelia Schmid, and Josef Sivic. Detecting unseen visual relations using analogies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1981–1990, 2019.
  - [15] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016.
  - [16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
  - [17] Hai Wang, Wei-shi Zheng, and Ling Yingbiao. Contextual heterogeneous graph network for human-object interaction detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 248–264. Springer, 2020.
  - [18] Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, Xiangyu Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4116–4125, 2020.
  - [19] Bingjie Xu, Yongkang Wong, Junnan Li, Qi Zhao, and Mohan S Kankanhalli. Learning to detect human-object interactions with knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
  - [20] Xubin Zhong, Changxing Ding, Xian Qu, and Dacheng Tao. Polysemy deciphering network for robust human–object interaction detection. *International Journal of Computer Vision*, 129:1910–1929, 2021.
  - [21] Penghao Zhou and Mingmin Chi. Relation parsing neural network for human-object interaction detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 843–851, 2019.
  - [22] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.