目　录

一种应对场景图生成任务的可端对端训练的模型

摘　要

场景图生成（Scene Graph Generation）任务旨在从图片中提取结构化表示，以得到对图像内容的完整描述。相比于物体检测而言，场景图生成任务不仅要求检测出图片中所有物体，还需要判断出哪些物体之间存在显著关系以及存在何种关系。近年来，许多方法搭建在基于 RPN 的物体检测器上，这些模型通常需要多阶段训练并且网络架构由多个模块堆叠形成，而如何搭建一种更加简洁的模型却需要进一步探索。与此同时，已经有成功应对物体检测任务和人物交互检测任务的端对端模型。本文将跟随他们的工作，探讨是否有可能建立针对场景图生成任务的端到端的模型。本文的主要工作如下：

（1） 搭建了以一个基于 Transformer 的端到端的场景图生成模型，我们称之为 SGGTR。

（2） 针对模型中由于成对组合而导致关系规模过高的问题，提出了使用交叉注意力机制将关系的复杂度从$O(n^2)$降到$O(n)$。

（3） 在公开数据集进行了大量实验，模型在 Visual Genome 数据集上达到了 28.9recall@100，在 HICO-DET 数据集上达到 21.94mAp，正在接近目前主流模型的性能。

关键词：场景图生成；端到端模型；视觉关系；人物关系检测；Transformer

ABSTRACT

Scene graph generation aims at extracting structural representation from image to get holistic description for image content. Compared with object detection, scene graph generation doesn't target at detecting all objects in an image, but also needs to determine whether there exists significant relationships between objects and if exists, what is the relationship's categories. In recent years, we have witnessed many robust models which are built upon RPN-based detector. However, these models usually need multi-stage training and their architectures are complex with stacked multi-stage modules. How to build a more compact and concise model needs to be explored further. At the same time, there already exists successful end-to-end models for object detection and human-object interaction. In this paper, we will follow their works to explore whether it is possible to build an end-to-end model for scene graph generation. The main works are concluded as follows:

(a) An end-to-end model for scene graph generation using Transformer is proposed which we call SGGTR.

(b) In order to solve the problem that the relationship's scale is too large due to pairwise combination, a solution using cross attention mechanism is proposed to reduce the complexity of the relationship from $O(n^2)$ to $O(n)$.

(c) Lots of experiments have be conducted. The model achieves 28.9 recall@100 in Visual Genome dataset and 21.94 mAp score in HICO-DET dataset which is approaching the state-of-the-art models' performance.

Keyword: Scene Graph Generation; End-to-end Model; Visual Relationship; Human-Object Interaction; Transformer

第 1 章 绪论

1.1 研究背景和意义

如同我们人类一样去感知环境和与我们周围的世界进行交互是人工智能的目标，为了实现这一美好愿景，计算机视觉在其中扮演了重要角色，至今，计算机视觉已经在感知图像方面取得了很大进步，但是在推理和认知方面和我们的预期存在着较大差距。为了填补这一空白，很多研究者开始聚焦在诸如图像标题自动生成、视觉问答、视觉知识推理等高层任务。而在这其中，场景图生成任务扮演着桥梁的角色来连接诸如目标检测底层任务和这些高层语义理解任务。

人类通过捕捉不同物体之间的关系，以获取对周围世界的整体性认知。我们并不满足于检测出图像中的单个物体，还侧重于分析它们是如何组织起来以共同完成对图像的内容描述。场景图首次在 2015 年被提出[1]，它提供了一种描述图像高级语义内容的逻辑表示方式。场景图是一种图结构模型，图中节点表示图像中的对象，节点关系表示物体之间的视觉关系。而场景图生成任务的目的是从图片中提取场景图，以得到对图像内容的整体描述。场景图生成任务是一种连接诸如物体检测底层任务到图片语义理解高层任务的桥梁，近几年的工作揭示了场景图对诸如图像检索[1]、目标检测[2]、视觉问答[3]、图片标题生成[4]等下游任务有益处。学术界针对场景图生成任务提出了很多方法，然而由于视觉层面的高度内在差异、数据集长尾分布[5]、数据集不完全注释[6]、语义歧义[7]等问题，它至今仍然是一个具有挑战性的课题。

1.2 国内外研究现状

现存方法大多搭建在主流物体检测网络之上，首先使用基于 RPN 的检测器来寻找所有可能存在物体的位置，从候选框中截取出特征，将其送入之后的网络来判别物体之间的关系和类别，并将这些检测到的对象和关系组装成场景图。然而，这样两阶段方法存在三个明显的局限性，首先，它们是以一种多阶段的方式处理图像，在训练期间，需要首先训练一个目标检测器或加载一个预训练物体检测模型，然后再训练关系检测分支。在推理期间，目标检测器将会生成数千个候选框，这些候选框可能组装成数以万计对关系，需要启发式方法对这些关系进行

评分的过滤。其次，这种两阶段的方法割裂了目标检测的视觉关系检测，关系检测很大程度上依赖于物体检测，如果一个与图片中很多物体存在关系的对象没有被检测出来，则后面的分支再也没有可能检测出与这个对象关联的所有关系。第三，如同绝大多数基于 RPN 的方法一样，两阶段方法通过 RoI 对齐的方式从特征图中提取物体和关系的视觉特征，这么做将导致一个问题，当不同类型的关系的区域高度重叠[8]，由于缺乏自适应选择特征的机制，这样简单地从目标区域裁剪出特征将会引入噪声，从而应发后续分类的歧义。

1.3 解决的主要问题

本文试图解决上述三个问题，对于第一个问题，我们将 CNN 和 Transformer 以端对端的方式堆叠在一起，形成一个简单而有效的网络。如图 2-1 所示，二阶段方法需要候选框提取和目标关系检测分类这两大步骤；单阶段算法将关键点检测思想应用到这里，网络结构简洁、漂亮，但是它需要人工构建的结构和复杂的后处理操作。而我们的方法是一种端到端的网络，将图查询和 CNN 特征输入到 Transformer 中，输出即为场景图，没有任何复杂的后处理操作。对于第二个问题，我们将目标检测分支和关系检测分支并行地输入到 Transformer 中，融合物体和关系的上下文，在 Transformer 的自注意力机制中，物体和物体、物体和关系、关系和关系之间的信息可以同时传递，形成一个更加灵活的推理过程。对于最后一个问题，我们使用交叉注意力机制从特征图中自适应地选择特征。

1.4 本文的主要工作

针对现存两阶段方法的不足，主要对如何构建一个应对场景图生成任务端到端的模型进行了探索，本文的主要研究工作如下：

（1） 提出一个基于 Transformer 的端到端的模型

在基于 Transformer 的子图推理模块中构建出图像二维网格点图、输入查询图、外部知识语义图谱并使用 Transformer 对视觉特征、语义特征和查询特征进行特征融合。使用交叉注意力机制和自注意力机制分别进行特征选择和消息传递。最后，设计分层损失来对已标注关系标签和未标注关系标签进行不同程度的惩罚，设计知识损失来拉近场景图中具有相关意义节点在语义空间的距离。

（2） 使用交叉注意力机制减少完全图中视觉关系的复杂度

针对完全图的密集关系进行优化，设计隐含关系查询向量，从大规模的关系中自适应地选择出存在实际含义的关系，降低了后续处理关系的复杂度。

（3） 在数据集上进行大量的实验

在两个主流数据集上测试了模型的性能，并和当前方法进行对比，进行了对模型结构和损失的消融实验。对实验结果进行可视化展示和分析。

1.5 论文的组织结构

第一章绪论，主要描述了场景图生成任务的背景及其意义，目前主流方法存在的问题以及本文如何解决该问题。

第二章相关研究工作，回顾目前在场景图生成任务、人物关系检测任务、目标检测任务上的主流方法，列举了Transformer在计算机视觉中的应用，说明本文方法的思想来源以及和前人方法的区别。

第三章主要方法，重点介绍针对场景图生成任务端到端模型的结构，并说明结构设计背后的动机。

第四章实验，介绍实验环境和所使用的数据集，对比本文方法和主流方法在数据集上的性能，并对结果进行分析。

第五章总结和展望，总结本文内容，指出未来研究方向。

第 2 章 相关研究工作

2.1 场景图生成

2.1.1 方法概述

如图 2-1 所示，目前针对场景图生成任务的方法可分成两种：单阶段方法和两阶段方法。现存的场景图生成方法[9][10][11][12][13][14]大多搭建在基于 RPN 的检测器上，这些方法首先利用候选框生成网络 RPN 产生可能存在物体的候选框，从每个候选框中提取视觉特征，然后利用这些视觉特征构建成图结构，并应用几轮图节点间消息传递迭代，最后在预测阶段通过从视觉特征、空间关系特征和语义关系特征中提取线索对关系进行分类。 单阶段方法与这些两阶段方法即先生成候选框再对关系进行分类的方法不同，FCSGG[15]受人体姿态估计方法的启发，将场景图生成任务视作关键点估计，并行地对物体和关系进行定位和分类，与两阶段方法相比，他们的模型结构简单且显著缩短了推理时间，但是他们的方法仍然存在人为干预。本文所提出的方法和这些方法不同，本文构建了一个真正端到端的网络，消除了诸如 RAFs[16]人工设计的组件以及复杂的后处理操作，网络结构非常简单，且网络输出即所得。

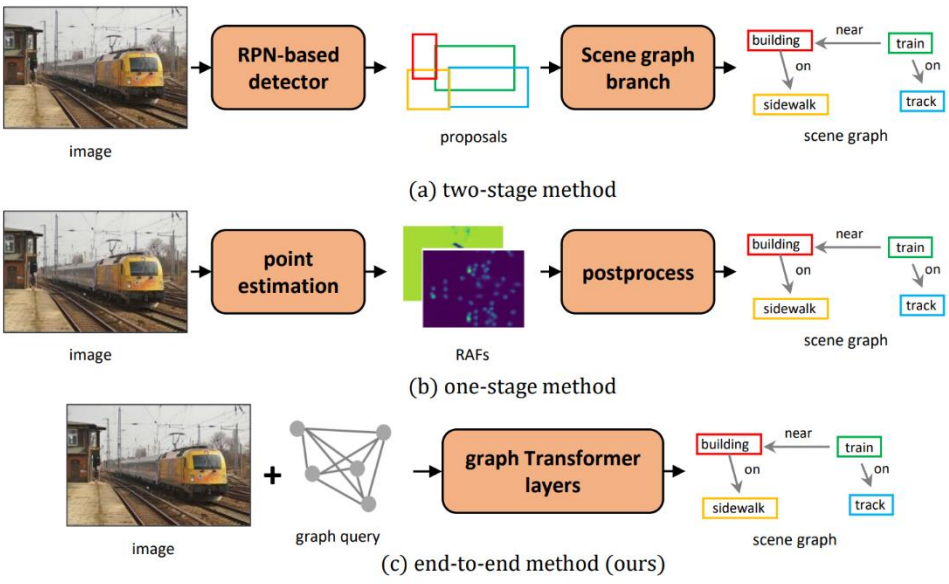


图 2-1 场景图生成任务三种不同类型的方法

2.1.2 结合上下文环境的方法

上下文环境在对判别场景图节点类别存在重要作用，图片中的物体能够依赖关系相互印证彼此的存在，很多工作受此启发，将各种图模型引入到场景图生成模型中来进行各个节点间的消息传递。Li Fei-Fei 组首次使用 GRU 来对图节点进行特征融合和消息传递[9]，后续的工作融入了更加复杂的模型，诸如 GCN，GNN，Transformer。这些工作一般是基于 RPN 的两阶段方法，使用 RPN 从图片找到候选框，从候选框中提取特征向量，以这些特征向量为图节点隐含向量，构建出图结构，在构建的图中进行使用消息传递，以达到特征融合的目的。受这些工作的启发，本文使用 Transformer 中的自注意力机制进行消息传递来融合上下文信息，但本文方法和这些方法[13][17][18]不同，我们使用 Transformer 不仅在视觉特征层面做特征融合，而是融合多源输入，在特征网格图、场景图查询图和由外部知识所构建的知识图谱做特征交换与融合。

2.1.3 融合外部知识的方法

我们人类在观察周围世界的同时，总是带着先验知识，先验知识的引入可以缓解模型对数据需求的压力和减少预测过程中产生的歧义。有两种外部知识，一种是从大规模语料库中学习得到的词嵌入，他们隐式反映了词与词之间的关系。第二种是由人为构造的知识图谱，在这个图谱中，容纳了世界万物之间的关系，比如说 man 和 woman 共同输入 person 这个子集[19]。Li Fei-Fei 组最早将语义先验引入到模型中[20]，后续的方法也使用语义先验来提高模型的鲁棒性，他们使用词嵌入表示向量构建出一个语义空间，并把视觉特征投影到对应的语义空间中，使这些向量能够在这个空间中满足语义关系，之后的工作也使用语义向量来作为视觉向量的补充。最近学术界更喜欢构建出一张独立于场景图的语义图，建立语义图和场景图之间的联系，从而在这两张图之间做推理。受这些工作的启发，我们也尝试将外部知识容纳进我们的模型中，我们跟随这个方法[19]使用词嵌入和人为构造的知识图谱，我们通过两种方式将外部知识融入到我们的模型之内，一种是通过设置损失的方式，设置额外的损失使特征能够投影在对应的语义子空间空间中，另外一种是通过图融合的方式，使用交叉注意力机制融合场景图和知识图。

2.2 人物关系检测

人物关系检测是和场景图生成任务十分相关的任务，人物关系检测目的在于从图片检测出<人，物，关系>三元组，与场景图生成任务不同的是，人物关系检测是生成以人为中心的场景图，而且图像中的关系并不像场景图生成任务那样密集。这两项任务上有很多相似之处，许多观点相通。人物关系检测的方法也可以分成三个分支：两阶段方法[21][22][23][24]、单阶段方法[25][26][27]和端到端的方法[28][29]。两阶段模型一般被设计成一种多流结构，从视觉流、语义流、空间关系流中预测关系的类别，这种结构和场景图生成任务中的两阶段算法面对相同的局限，即依赖于提前训练好的物体检测器。单阶段方法，模型结构简单且高效，通过预测关键点来估计人和物的位置，使用关系中间节点来关联人和物，但是存在单阶段方法所共同面对的局限，即需要人为构造的结构和复杂的后处理操作。实际上，针对人物关系检测已经存在端到端的模型，这些方法以 DETR 结构为基础，输入、输出有改动，输入查询向量，输出的是以三元组为目标的特征，特征被用来对物体、人的位置回归和物体、关系类别预测，他们的方法是针对人物关系三元组的，我们的方法虽然也使用到 DETR 结构，但是我们的方法可以覆盖更一般的情形，输入和输出是一个图结构，而不是对单个关系进行预测。

2.3 目标检测

场景图生成任务总是伴随着目标检测的方法而不断更新、迭代。目标检测可分成两阶段方法、单阶段方法和端到端方法。Faster R-CNN 是一种典型的两阶段方法，它在第一阶段生成候选框，在第二阶段对这些候选框进行分类和选择过滤。场景图生成任务大都采用 Faster R-CNN 的结构，不同的是把第二阶段的物体分类改成了物体和关系的分类。近一两年来，学术界在追求单阶段算法和消除非极大值抑制的端对端算法，CenterNet 是一个典型的单阶段方法，它使用点估计的方法合并了回归和分类，FCSGG[15]受此启发使用它和构建出针对场景图生成任务的单阶段方法。DETR 是一种端到端的物体检测方法，它结构简单且消除了诸如非极大值抑制的后处理操作。本文将跟随他们的工作，构建出一个针对场景图生成任务的端到端模型。

2.4 卷积之外的图推理

卷积由于其感受野的局部性很难建立长程联系，而像语义分割这样的任务又需要捕获物体内在各个部分的依赖性，为了捕获远程依赖关系和上下文环境，一些工作[32][33][34][35]尝试将图模型集成到他们的网络中，它们采用（1）从图像特征空间投影到语义符号空间，（2）在语义符号空间做符号推理，（3）从语义符号空间反投射到图像空间以细化原有特征三个步骤来做卷积操作之外的图推理。本文的方法也将这三个步骤结合到我们的模型中，但与使用 RPN 来限制特征区域的方法[36]不同，本文使用交叉注意力机制来自适应的选择性投影和反映射视觉特征。

2.5 Transformer 在计算机视觉的应用

Transformer[38]曾经在机器翻译中取得的很好的效果，近两年来，大量的工作尝试将 Transformer 应用到计算机视觉中的各个任务中，比如多模态学习[39][40][41]、图像分类[42]、目标检测[43]、语义分割[44]等。Transformer 相比于传统 CNN 的最大优势在于它可以捕获长程联系，一些工作证明 Transformer 在某些地方可以达到与 CNN 相同的性能，但是由于 Transformer 中注意力层中的计算复杂度是$O(n^2)$，将图像作为输入，复杂度将会达到不可接受，之后的一些工作也探索了如何降低 Transformer 的计算复杂度，本文的模型也面临相同的问题，即输入序列过长会导致 Transformer 层复杂度过高的问题，本文采用的方法是使用 CNN 对图像下采样到用来的 32 分之一，并固定查询序列的数量在 200 之内，以这样的方式有效降低了 Transformer 在计算注意力权重时的复杂度。虽然已经存在将 Transformer 应用于场景图生成任务或视觉关系检测任务的方法[13][17][18]，然而，他们的方法使用 Transformer 来融合图节点之间的特征，并且他们的方法仍然是基于 RPN 的两阶段方法。我们的工作与他们不同，我们将跟随 DETR[43]去搭建一个真正端到端的模型，使用 Transformer 在三张图中特征融合和推理，而且其中的检测分支是基于编码器-解码器的结构，而不是基于 RPN。

第 3 章 主要方法

3.1 方法概述

如图 3-1 所示，我们的模型有三种类型的输入：（1）视觉特征序列，（2）场景图查询，（3）外部知识。将它们输入到由多层 Transformer 子层堆叠形成的子图融合模块，使得查询能够使用交叉注意力机制从视觉特征分支和外部知识分支提取场景图特征信息。我们的 Transformer 子层被设计成一种级联结构，逐步产生场景图。在查询结果几轮特征聚集之后，会形成场景图特征序列，这些序列将会被去回归物体的位置和分类物体、关系的类别，从而可以构建出场景图。

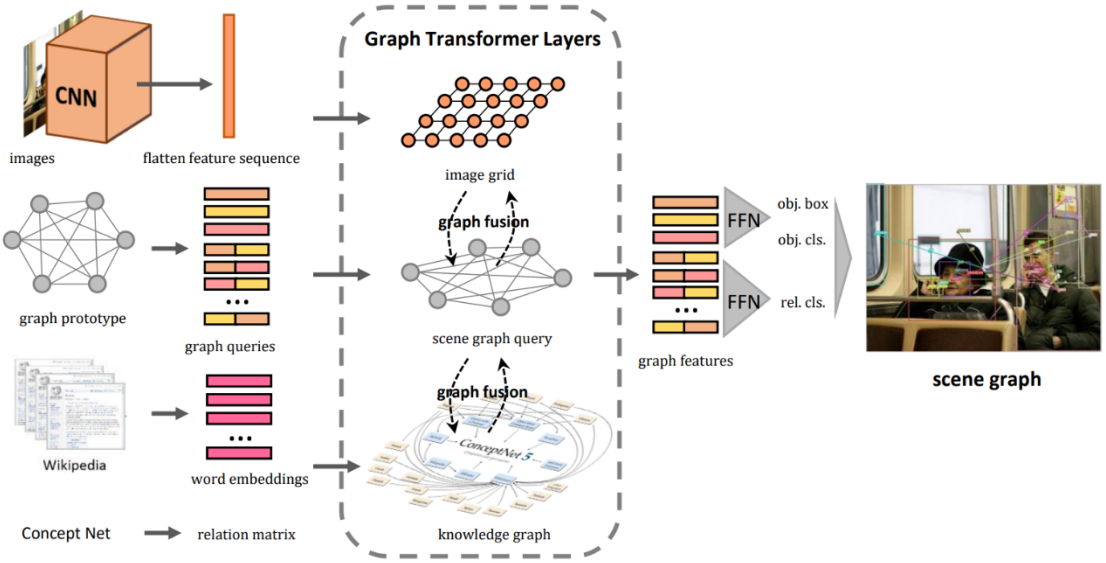


图 3-1 SGGTR 总体架构

3.2 多源输入

3.2.1 视觉分支

给定形状为$\mathbb{R}^{w\times h\times 3}$的一张图片$I$，我们使用 CNN 去提取特征，得到 2D 特征图$F\in\mathbb{R}^{\frac{w}{s}\times\frac{h}{s}\times c}$，这里$s$是 CNN 的卷积总步长，$c$是输出特征图的维度。由于 Transformer 只能接受 1D 序列的输入，我们像大多数方法一样，将二维特征图展开成长度为 n 的一维特征序列$\{\mathbf{f}_i\}_{i=1}^{n}$。为了保留二维空间位置信息，我们仿照 DETR[43]使用由 sine 函数计算的位置嵌入[45]。

3.2.2 外部知识分支

本文跟随 GB-Net[19]使用词嵌入向量表示和先验关系来构建我们的知识图谱，词嵌入表示是从 GloVE 根据数据集中所有物体和关系的类别选择出来的，而先验关系是从 Concept Net 收集得到的。从形式上讲，将由外部知识所构建出的图记为$\mathcal{G}^S = \{\mathcal{V}^S, \mathcal{E}^S\}$。它有两个组成部分，即图节点集合$\mathcal{V}^S$和关系集合$\mathcal{E}^S$。图节点涵盖了数据集中的所有类别，拿 Visual Genome 数据集举例，该数据集中存在 150 个物体类别，那么该图中就存在 151 个物体图节点，新增的一个节点表示空物体，类似的还有 51 个关系图节点。这些节点使用词嵌入表示向量作为自己的语义特征。图中的边表示图节点以何种方式关联，这些关联方式在我们的实现中表示成不同层次的关系矩阵，存在联系的节点对在关系矩阵中对应位置置 1，否则置 0，我们推荐读者去 GB-Net[19]查看更多的细节。

3.2.3 图查询分支

查询图$\mathcal{G}^Q$是一个图，图中节点起初没有特定含义，这张图只表示初始场景图的拓扑结构，我们将这张图称之为场景图原型。这个场景图原型可以被视为是一种带有记忆功能的图，它保存了各种各样的场景图原型，就如同我们的大脑，即使我们闭上眼睛，我们的大脑仍然可以浮现出各种场景。我们固定了 100 个物体查询，并将所有对象成对连接起来，形成一个完全图。在这里，成对组合物体对于搭建一个真正端到端的模型是必要的，我们没有采用像 HOTR[30]这样的更加复杂的关系查询，因为它会在推理中引入更复杂的后处理过程并且在训练阶段引入更复杂的损失计算。像 DETR[43]一样，我们将所有图查询初始化为一个零向量，并对每个查询附加一个在训练过程中学习得到的位置向量以区分图中的各个节点和保留拓扑结构。对于物体q_i^o，它学习得到的位置向量表示成p_i^o，对于物体查询节点q_i^o和物体查询节点q_j^o，位置向量通过下式计算

$$p_{ij}^r = \mathrm{FC}\left(\mathrm{Concat}\left(p_i^o, p_j^o\right)\right) \tag{3-1}$$

我们连接两个物体位置向量并使用全连接层使他们和物体位置向量的维度保持一致。

3.3 Transformer 图融合模块

Transformer 图融合模块是由几个堆叠的 Transformer 子层构成，它的输入包括视觉特征序列$\{\mathbf{f}_i\}_{i=1}^{n}$、场景图查询$\{\mathbf{q}_i\}_{i=1}^{m}$和带有语义信息的词嵌入表示向量序列$\{\mathbf{s}_i\}_{i=1}^{k}$，输出是场景图的特征。Transformer 模块是以一种级联的方式处理这些输入，如下式所示，

$$\{\mathbf{q}_i^{l+1}\}_{i=1}^{m} = \mathcal{G}\left(\{\mathbf{f}_i^{l}\}_{i=1}^{n}, \{\mathbf{q}_i^{l}\}_{i=1}^{m}, \{\mathbf{s}_i^{l}\}_{i=1}^{k}\right), l = 0,1,2,\dots,N \tag{3-2}$$

这里上标表示该向量序列所属的层级，N是 Transformer 子层的数目，$\mathcal{G}$表示 Transformer 模块中的一个子层，将在下面详细展开对它的介绍。

在 Transformer 子层中，使用交叉注意力机制在三个来源不同的输入做特征融合。如图 3-2(a)所示，信息流通包括四个方向：（1）从视觉分支流入查询图分支，（2）从查询图分支流入知识图谱分支，（3）从知识图谱分支流回查询图分支，（4）从查询图分支流回视觉特征分支。首先，我们使用交叉注意力机制让查询图向量序列$\{\mathbf{q}_i\}_{i=1}^{m}$去从视觉特征分支中聚集视觉信息。

$$\{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m} = \mathrm{CrossAttention}(\{\mathbf{q}_i\}_{i=1}^{m}, \{\mathbf{f}_i\}_{i=1}^{n}, \{\mathbf{f}_i\}_{i=1}^{n}) \tag{3-3}$$

这里$\{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m}$是一个展开来的场景图视觉特征，$\mathrm{CrossAttention}(Q,K,V)$是一个以$Q$为查询、以$K$为键、以$V$为值的多头注意力层。在特征信息从视觉分支流入场景图分支后，场景图查询中的每个向量有了它相对应的视觉含义。然后，我们把这些场景图视觉特征投影到使用先验知识构建的语义图中，

$$\{\mathbf{f}_i^{Q\to S}\}_{i=1}^{k} = \mathrm{CrossAttention}\left(\{\mathbf{s}_i\}_{i=1}^{k}, \{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m}, \{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m}\right) \tag{3-4}$$

这里$\{\mathbf{s}_i\}_{i=1}^{k}$是语义知识图谱中的节点的词嵌入表示向量，$\mathbf{f}_i^{Q\to S}$是语义图谱中融合视觉特征后的结点$i$的特征。正如第 3.2.2 节所示，这张语义图谱中的图节点由词嵌入表示向量组成，图中的边由两个关系的先验知识构成。在我们将带有视觉信息的场景图向量投影到该语义图谱中之后，我们希望在图片中出现的那些物体和关系在这张语义知识图谱中能够被激活，就如同我们人类看到一张图片后，图片中的内容将会在我们大脑中形成各种印象和概念。在这种感知过程之后，我们还会结合我们头脑中的先验知识来分析图片中的物体是如何组织起来的。为了模拟这个过程，我们使用自注意力机制来在这张语义图谱中做图推导。

$$\{\mathbf{f}_i^{S\to S}\}_{i=1}^{k} = \text{SelfAttention}\left(\{\mathbf{f}_i^{Q\to S}\}_{i=1}^{k}, \text{weight} = M\right) \quad \textbf{(3-5)}$$

这里$\mathbf{f}_i^{S\to S}$是知识图谱中的节点i在自注意力机制过程从其他图节点收集特征之后的语义特征。我们使用注意力权重矩阵M来限制这张语义图上的信息传递过程，下面我们将展开介绍权重矩阵M的细节。注意到在这张语义图上，图节点表示概念，即数据集中表示物体的单词和表示关系的单词，这些图节点可以被分成m_o个物体图节点和m_r个关系图节点。拿 Visual Genome 数据集来说，我们将会有 202 个图节点，其中 151 个是物体图节点，51 个是关系图节点。正如前面所述的那样，语义图谱中的每个节点将词嵌入表示向量作为自己的语义特征，而注意力权重矩阵$M = (M_1, M_2, \ldots, M_k)$是一系列关系矩阵，每个关系矩阵是一个二值矩阵，它表明这两个概念即单词之间是否相关而且以何种方式相关。这里有k个关系矩阵，每个矩阵都代表不同的含义，诸如 ISA 关系、USEFOR 关系、SUBSETOF 关系等等，这个矩阵是由 GB-Net[19]构建的，我们建议读者去 GB-Net[19]查看更多的细节。如方程 3-5 表示的那样，我们将这些表示两个单词之间是否存在关系的注意力权重矩阵应用到自注意力权重上面。在这背后的动机很简单，我们希望网络可以关联那些意义相关的事物，就如同我们在看到“racket”之后，脑海里会浮现出“tennis”、“player”、“hold”、“short”等概念。在这个过程之后，语义特征将被融入视觉特征和和它相关的概念特征。这些语义特征将会被反投射到查询图中。

$$\{\mathbf{f}_i^{S\to Q}\}_{i=1}^{m} = \text{CrossAttention}\left(\{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m}, \{\mathbf{f}_i^{S\to S}\}_{i=1}^{k}, \{\mathbf{f}_i^{S\to S}\}_{i=1}^{k}\right) \quad \textbf{(3-6)}$$

至今，已经从另外两个分支收集到足够信息，初始为空的场景图查询向量将被赋予有意义的特征，这些特征取自于视觉图像和外部知识。为了去把握图片中物体与物体之间、物体和关系之间、关系和关系之间的依赖性，我们将自注意力机制应用到查询图中进行图推导

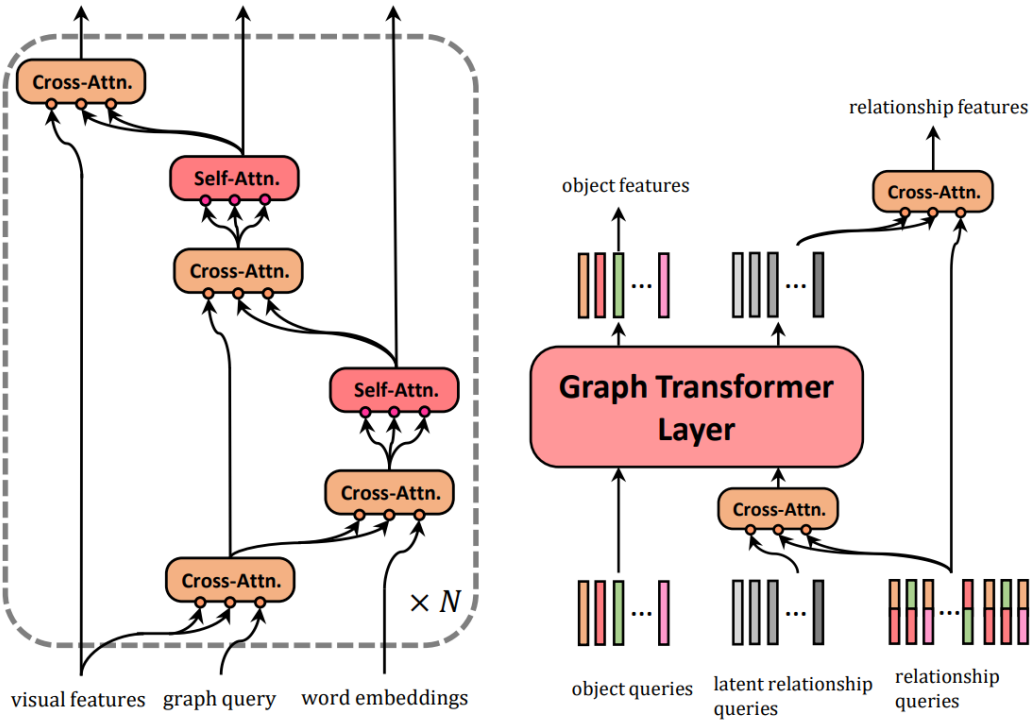
$$\{\mathbf{f}_i^{Q\to Q}\}_{i=1}^{m} = \text{SelfAttention}\left(\{\mathbf{f}_i^{S\to Q}\}_{i=1}^{m}\right) \quad \textbf{(3-7)}$$

这里$\{\mathbf{f}_i^{Q\to Q}\}_{i=1}^{m}$对应于方程 3-2 的$\{q_i^{l+1}\}_{i=1}^{m}$，在这个过程中，同一张图片中的物体和关系可以相互传递特征，并且上下文信息可以被散布到整张图的各个角落。查询图特征将会被用来去进行分类和回归，这将在章节 3.5 介绍。

最后我们将图查询特征反投影到视觉特征中去修正原先的视觉特征，

$$\{\mathbf{f}_i^{Q\to V}\}_{i=1}^{n} = \text{CrossAttention}\left(\{\mathbf{f}_i\}_{i=1}^{n}, \{\mathbf{f}_i^{Q\to Q}\}_{i=1}^{m}, \{\mathbf{f}_i^{Q\to Q}\}_{i=1}^{m}\right) \quad (3\text{-}8)$$

这里$\{\mathbf{f}_i^{Q\to V}\}_{i=1}^{n}$对应于方程 3-2 的$\{\mathbf{f}_i^{l}\}_{i=1}^{n}$。



(a) Graph Transformer layer (b) Graph query branch

图 3-2 Transformer 子层结构细节

3.4 减少完全查询图中关系的规模

在查询图分支，我们构建了一张完全图作为我们查询图的输入，列举了场景图中所可能出现的所有关系，当物体查询的数量增加时，关系查询的规模将会大到不可接受。同时我们观察到数据集中的关系数的规模并没有如此高，图片中并不是所有对象均存在关系。为了解决这个问题，即如何把密集完全图转换成稀疏图以减低后续计算复杂度，我们使用交叉注意力机制来减少关系的规模。如图 3-2(b)所示，我们定义了大小固定的隐含关系查询，这些隐含关系查询被初始化成零向量并且他们位置向量表示在训练中通过学习得到。隐含关系查询向量被输入进交叉注意力层中去从完全图中选择可能有具体含义的关系查询，之后的任何复杂操作，均附加在这些隐含关系查询向量中。最后这些隐含关系查询将会被反投影到原来的关系查询向量中。以这样的方式，我们把关系查询的复杂度从$O(n^2)$降到$O(n)$，与此同时没有破坏完全图结构。

3.5 推理和损失

3.5.1 推理

在场景图查询从视觉分支和语义分支收集到足够多的特征之后，这些特征将会被用于分类和回归。对于物体查询来说，使用单 MLP 层分类，使用三隐层 MLP 回归，即

$$\mathbf{b}_i = \mathrm{MLP}\left(\mathbf{f}_i^{Q\to Q}\right) \text{ and } c_i^o = \mathrm{MLP}\left(\mathbf{f}_i^{Q\to Q}\right) \tag{3-9}$$

这里$b_i \in \mathbb{R}^4, c_i^o \in \mathbb{R}^{C_o+1}$，$C_o$是数据中所有物体类别的数量。对于关系查询来说，他们仅仅被用于分类，由于和关系相关联的物体已经隐含在关系的位置向量上，这里不必再去对关系的位置做回归，

$$c_{ij}^r = \mathrm{MLP}\left(\mathbf{f}_{ij}^{Q\to Q}\right) \tag{3-10}$$

这里$c_{ij}^r \in \mathbb{R}^{C_r+1}$是物体$i$和物体$j$之间的关系，$C_r$是数据集中所有关系类别的数量。在推理阶段中，首先把空物体查询排除，并且只保留与关系关联的两个物体均不为空的关系，利用这些物体的位置、类别和关系的类别构建出场景图，注意到这里没有引入任何复杂的后处理步骤。

3.5.2 损失函数

在训练阶段我们使用 Hungarian 二分匹配[43]来将标记赋给每一个预测，损失通过下式计算

$$\mathcal{L}_{\text{total}} = \lambda_{\text{L1}}\mathcal{L}_{\text{L1}} + \lambda_{\text{IOU}}\mathcal{L}_{\text{IOU}} + \lambda_{\text{CLS}}\mathcal{L}_{\text{CLS}} + \lambda_{\text{knowledge}}\mathcal{L}_{\text{knowledge}} \tag{3-11}$$

这里$\mathcal{L}_{L1} = \sum_i \left\|\mathbf{b}_i - \hat{\mathbf{b}}_i\right\|_1$是预测候选框$\mathbf{b}_i$和标记候选框$\hat{\mathbf{b}}_i$之间的$l_1$损失，$\mathcal{L}_{\text{IOU}}$是广式 IoU 损失。我们使用交叉熵损失来计算$\mathcal{L}_{\text{CLS}}$。为了避免模型过于倾向于把查询特征分类成空物体和空关系，我们对空关系和空物体设置了一个权重。与 DETR[43]相比，我们新增了两个损失，一个用来计算关系分类的损失，另外一个损失用来对知识投影进行惩罚。

我们把预测关系分支的输出视作一个大小为$\left(n_{obj}, n_{obj}\right)$的二维矩阵，这里$n_{\text{obj}}$表示物体查询的数目。在这个矩阵中，位置$(i,j)$是一个逻辑概率向量$c_{ij}^r \in$

$\mathbb{R}^{C_r+1}$，它代表这物体i和物体j之间的关系被分成各类的概率。我们根据标记和预测结果二分匹配的结果来构建真值矩阵，这个真值矩阵中的每一个位置(i,j)被赋值成图查询节点i和图查询节点j所对应的物体之间的关系标签。构建完成这两个矩阵之后使用带有对空关系的惩罚权重为 no_rel_weight 的交叉熵损失来计算关系的损失。为了进一步区分已标记标签和未标记标签，我们对关系设置了两个损失，第一个损失就如同上述的那样使用权重为 no_rel_weight 的交叉熵损失来计算预测关系矩阵和真值关系矩阵之间的差距。我们还添加了对已标记标签的辅助损失，对于那些数据集已经标注的标签，使用权重 aux_no_rel_weight 来计算交叉熵损失。

至于知识投影损失，首先把物体特征和关系特征使用单 MLP 层投影到一系列具有不同含义的隐语义空间中。在每个语义空间中，每一个物体和关系都被赋予一个语义标签，这个语义标签由外部知识所定义。例如，在表示“SIMILAR WITH”含义的隐语义中，由于在外部知识中“wearing”和“has”、“wears”表达同一个含义，那么标签为“wearing”相对应的查询被分类成“has”和“wears”多标记。在这背后的动机是，我们希望带有相同含义的单词应该被投影在相同的语义空间中。

第 4 章 实验

4.1 数据集

实验中我们使用 Visual Genome 数据集[46]来测试场景图生成的性能，使用 HICO-DET[47]数据集来测试视觉关系检测的性能。

4.1.1 Visual Genome 数据集

Visual Genome 数据集[46]是一种验证场景图生成模型性能的主流数据集，这种数据集标注了图片中物体的类别、包围框和物体之间的关系，这些关系可以分成空间位置关系、从属关系、语义关系，这些关系中很大一部分是以人为中心的关系，例如“man-wears-pants”、“player-holds-racket”，位置关系主要有“on”、“near”、“above”等，但语义关系较少。在实验中，我们也观察到场景图有很大一部分在描述某个组件的组成部分，例如“bus”和“wheel”、“windows”、“sign”之间的关系，“person”和“head”、“leg”、“hair”之间的关系，这些以单个物体为中心的关系给我们设计后续的模型结构有很大的启发。Visual Genome 数据集包含了 108,077 张图片，由于原数据集的标注存在噪音且分布十分不均，我们在实验中采用了比较流行的划分方式[48]，这种划分方式从数据集中选取物体和关系出现次数最多的 150 个物体类别和 50 个关系类别。

由于数据集自身的不完全标注问题，使用准确度测评模型将会导致效果不准，在这个数据集上主要的评测指标为召回率。将模型的输入组织为一个包含主体-关系-客体三元组的列表，按照三元组的概率对该列表进行排序，预测的关系列表和真值关系列表之间计算 recall 值，每一张图片均要计算 recall 值，最后取平均。考虑到同一种关系可以有不同的标记，例如“wearing”和“has”，我们也仿照前人的工作，计算带有图限制的召回率和不带图限制的召回率。在带有图限制的召回率中，每一个关系只能被赋予一个标签，在预测的关系列表中，两个物体之间的关系三元组只能出现一次，而在不带图限制的召回率中，每一个关系可以被赋予多种标签，在这样的情况下，“girl-wears-shorts”、“girl-

wearing-shorts”和“girl-has-shorts”均可以出现在预测列表中，以这种方式计算出来的召回率会比带有图限制计算出来的召回率要略高。

另外，我们还对每一个关系类别计算它的召回率，类别判断正确的关系视为正例否则视为负例，按照这样的方式计算召回率。对每一类别的召回率取平均便可得到平均召回率 m-Recall，m-Recall 值相比于 Recall 值对模型的偏倚预测更加敏感，如果模型只对某一类的预测召回率较高，而对其他类别较低的话，m-Recall 会大受折损，所以 m-Recall 值可以反映模型是否存在偏倚问题。类似的平均召回率也可以分成带有图限制的平均召回率和不带图限制的平均召回率，它的计算方式和召回率计算方式相同。

4.1.2 HICO-DET 数据集

HICO-DET 数据集[47]是针对人物关系检测的数据集，数据集中标注了人物之间的交互关系，诸如“person-washing-car”、“person-board_on-plane”、“person-ride-bike”等关系。不像 Visual Genome 数据集那样，一张图片中包含大量关系，HICO-DET 数据集中的图片中关系数目较少，也存在单人多物交互、单物多人交互和多人多物交互。数据集对人和物体和人物之间的交互做出了位置和类别标注，和 Visual Genome 不同的地方在于，HICO-DET 数据集中不同关系中的人物的标注框不完全一致，同一个人在不同的关系中存在不同的候选框标注。数据集中的关系大多存在语义关系，例如“washing”、“inspect”等，这也是这个数据集的难点之一。

HICO-DET 数据集包含了 47,776 张图片，其中 38,118 张图片用于训练，9,658 用于测试，包含 71 和类别和 117 个交互关系类别。该数据集的标注比较完整，虽然也存在一些未标注的数据，但是占比很小，该数据集上的评测指标为 mAp，mAp 的计算方法和物体检测计算 mAp 方式类似，模型输出一个列表，列表中每一项是人物关系三元组，列表按照三元组的概率排序，这个列表和真值标注对比，人、物和人物之间的关系的类别正确，并且人、物的候选框和真值标注的候选框之间的 IoU 大于 0.5，只有满足以上情况才属于正例，其他情况属于负例。每个类均计算 Ap 值，最后去平均得到 mAp 值。根据数据集中关系类别出现的频率分成 rare 和 non-rare 两种，并分别计算出现次数较少的关系的 mAp 值，出现次

数正常的 mAp 值和所有关系的 mAp 值。

4.2 模型实现细节

跟随 DETR[43]使用 ResNet-50 来提取特征，特征图由 ResNet-50 输出且输出特征的通道数为 2048，我们将它下采样到 256。我们固定了维度为 256 的 100 个物体查询和 100 个隐含关系查询。在 Transformer 层中，使用 5 层 Transformer 子层，每一层使用 8 个并行的多头注意力层，前馈特征的维度为 2048。在损失函数中，设置λ_{L1}为 5，λ_{IOU}为 2 并且λ_{CLS}为 1，空物体权重设置为 0.1，空关系权重设置为 0.01。使用 AdamW 优化器来训练模型，初始学习率设置为5×10^{-5}，并且在 60 轮训练中衰减为原来的 0.1。训练之初加载 ResNet-50 并冻结批归一化层，我们使用 4 卡进行训练，批大小设置成 8，整个训练过程大概消耗 4 天。

表 4-1 SGGTR 在 Visual Genome 数据集上和其他模型的性能对比

Recall @ K / No-graph Constraint Recall@K		Scene Graph Detection											
		R@20/50/100			ng-R@20/50/100			mR@20/50/100			ng-mR@20/50/100		
External Knowledge	VCTree[51]	22.0	27.9	31.3				5.2	6.9	8.0			
	KERN[52]		27.1	29.8		30.9	35.8		6.4	7.3			
	GPS-NET[53]	22.3	28.9	33.2						9.8			
	MOTIFS-TDE[50][54]	12.4	16.9	20.3				5.8	8.2	9.8			
	GB-NET[19]		26.3	29.9		29.3	35.0		7.1	8.5		11.7	16.6
	RelDN[56]	21.1	28.3	32.7		30.4	36.7						
Visual Only	VTransE[57]		5.5	6.0									
	FactorizableNet[58]		13.1	16.5									
	IMP[48][50]	14.6	20.7	24.5									
	Pixels2Graphs[59]					9.7	11.3						
	Graph R-CNN[49]		11.4	13.7									
	VRF[60]		13.2	13.5									
	CISC[36]	7.7	11.4	13.9									
	HRNet[15]	16.1	21.3	25.1	16.7	23.5	29.2	2.7	3.6	4.2	3.8	5.7	7.5
	CMAT[15]	22.1	27.9	31.2	23.7	31.6	36.8						
	KERN[14]		27.1	29.8		30.9	35.8		6.4	7.3			
	LSBR[61]	23.6	28.2	31.4	26.9	31.4	36.5						
	SGGTR (Ours)	18.7	24.7	28.9	20.5	28.1	33.5	4.0	5.9	7.4	5.8	9.7	13.7

4.3 结果

我们收集了近四年在 VG 数据集和 HICO-DET 数据集上的评测结果，如表

4-1 和表 4-2 所示，我们的结果正在接近主流方法的性能，并且仍然存在不可忽视的差距。

在 Visual Genome 数据集上，如表 4-1 所示，我们跟随 FCSGG[15]把近几年的方法划分成两大类，一类是仅使用了视觉信息，另外一类处理视觉信息还使用到外部知识，在 FCSGG[15]中搜集的方法上，我们又新增了几个效果比较好的且近几年刚刚发表的方法，表格中列举了各个模型在 Visual Genome 数据集上的带有图限制的召回率、不带图限制的召回率、带有图限制的平均召回率和不带图限制的平均召回率，每种召回率列举前 20、50、100 召回率。我们的模型达到了 18.7 recall@20 并且打败了 2019 年之前的方法，诸如 Graph R-CNN、CISC、IMP。但是和最近发表的方法仍然存在差距，目前在 Visual Genome 数据集上最好的效果可以达到 30+的 recall@100，而我们的方法和他们的方法还存在几个点的差距。在 HICO-DET 数据集上，我们重点收集了 2021 年在 CVPR 会议上发表的方法，并把这些方法划分成单阶段方法、两阶段方法和端对端方法，每种方法列举了他们在出现频率不同层次的关系上的 mAp 值。表中最后一行显示了我们的结果，我们的结果和目前主流方法相比有几个 mAp 值的下降，尤其和最近刚刚发表的端到端的方法还存在不小的差距。

表 4-2 SGGTR 在 HICO-DET 数据集上和其他模型的性能对比

Method		mAp		
		full	rare	non-rare
Two-stage	CHGN[63]	17.57	16.85	17.78
	DRG[64]	24.53	19.47	26.04
	VCL[23]	23.63	17.21	25.55
	ATL[64]	23.81	17.43	24.32
One-stage	PPDM[25]	21.94	13.97	24.32
	IP-Net[26]	19.56	12.79	21.58
	GGNet[27]	23.47	16.48	25.60
	AS-Net[31]	28.87	24.25	30.25
	HOTR[30]	25.10	17.34	27.42
End-to-end	HoiTransformer[38]	26.61	19.15	28.84
	QPIC[29]	29.90	23.92	31.69
	SGGTR(ours)	21.94	17.14	23.37

4.4 消融实验

4.4.1 Visual Genome 数据集上的消融实验

在 Visual Genome 数据集上我们测试了三种模型结构，分别是 DETR-based SGGTR、SGGTR 和 SGGTR with knowledge branch。

（1） DETR-based SGGTR 是一种基于 encoder-decoder 架构的模型，它和原 DETR 的结构基本相同，不同的地方在于，输入由物体查询改成图查询并且加入了在 3.4 节所述的减小图查询中关系数规模的结构。

（2） SGGTR 是一个精简版的结构，它只包含视觉分支，没有加入外部知识。

（3） SGGTR with exknowledge 是我们完整的模型，它包含完整的三块输入。

每个模型在训练过程中配有不同的损失，由于物体候选框、物体类别和关系的类别的损失是模型训练过程所必须的，但外部知识损失和辅助的关系损失是附加的，我们只对后面两个损失展开消融实验。在进行实验过程中，为了加快模型收敛，而且也考虑到数据集中的图片足够多，训练过程中并没有使用数据增强。在数据集上训练一个完整的模型需要花费至少 4 天的时间，为了减少这些时间开销，我们训练过程会加载上一次训练好的模型参数，即相同的模型在不同的损失上进行微调。这些模型的结果见表 4-3 所示，表中 mAp 是物体检测的性能，recall@20/50/100 是 4.1 节描述的那样，我们分别随机从测试集和训练集中抽取 5000 张图片测试模型的性能。

表 4-3 SGGTR 在 Visual Genome 数据集上的消融实验

Model	w/o knowledge loss	w/o auxiliary relationship loss	on eval set				on training set			
				recall@				recall@		
			mAp	20	50	100	mAp	20	50	100
DETR-Based	✗	✗	23.61	13.47	20.93	25.90	53.71	31.99	43.69	51.87
SGGTR	✗	✗	24.50	18.45	24.30	28.56	34.46	24.56	29.96	31.24
	✗	✓	24.11	11.05	16.94	22.01	36.39	39.07	48.63	54.98
	✓	✓	22.32	10.01	15.20	20.23	32.78	35.88	44.20	49.64
SGGTR with exknowledge	✗	✗	22.36	14.66	21.82	26.28				
	✓	✓	21.99	9.96	15.35	20.19	43.53	41.75	52.06	58.60

对比模型 DETR-based SGGTR 和 SGGTR，SGGTR 在验证集上要略强于 DETR-

based SGGTR，这说明基于投影-推理-反投影的结构在处理场景图生成任务中要优于基于编码解码器的 DETR-based SGGTR，我们也发现，DETR-based SGGTR 在训练集上的物体检测效果要远高于其他模型，这说明基于编码解码器的结构对于物体检测任务是重要的。当 SGGTR 模型添加了辅助的关系损失后，模型在验证集上的性能大受损失，但是在训练集上的效果明显提高，这可能是由于过拟合导致的，注意到在训练过程中没有使用任何数据增强策略，这可能是导致 SGGTR 在添加了辅助关系损失后下降的主要原因。在添加了知识损失之后，模型在训练集和验证集的效果均会变差，这说明知识损失会扰乱之前的特征。注意到最后一行，带有外部知识的 SGGTR 模型虽然在验证集上的性能不如只含有视觉分支的 SGGTR，但是在训练集上却远远高于其他模型。这再次证明了包含三个完整分支的模型具有强拟合数据的能力，但是由于训练过程中没有使用数据增强操作而导致模型性能在验证集上大打折扣。

表 4-4 SGGTR 在 HICO-DET 数据集上的消融实验

Model	Object query number	Relationship query number	Graph Transformer layer number	mAp		
				full	rare	non-rare
DETR	20	20	5	22.31	18.36	23.49
SGGTR	20	20	5	21.93	18.45	22.97
	20	20	6	20.51	15.68	21.96
	50	50	5	21.94	17.14	23.37

4.4.2 HICO-DET 数据集上的消融实验

HICO-DET 数据集中，一张图片上存在的物体和关系数目较少，模型中查询图的规模不需要太大，由此我们对模型的 Transformer 层数和查询图的规模展开了消融实验。如表 4-4 所示，我们训练了两种模型，即 DETR 和 SGGTR，这里的 DETR 和在 4.4.1 节所述的 DETR-based SGGTR 的结构不同，表中的 DETR 完全照搬了原论文的结构，只是把输入更换成图查询，输出在对物体回归和分类的基础之上多增添了对关系的分类，而且这里的 DETR 模型也没有使用减少完全图中关系数的交叉注意力机制。由于目前没有方法在 HICO-DET 数据集上收集和组建外部知识，这里的 SGGTR 只包含视觉分支，不包含外部知识的输入。对比表中的第一行和第二行数据，我们发现 DETR 要在 HICO-DET 数据集上优于

SGGTR，SGGTR 性能稍有损失。而且从数据整体分布情况上来看，模型对查询图的规模和 Transformer 子层的层数并不敏感，限制模型性能可能是模型的结构和损失的设置不合理。

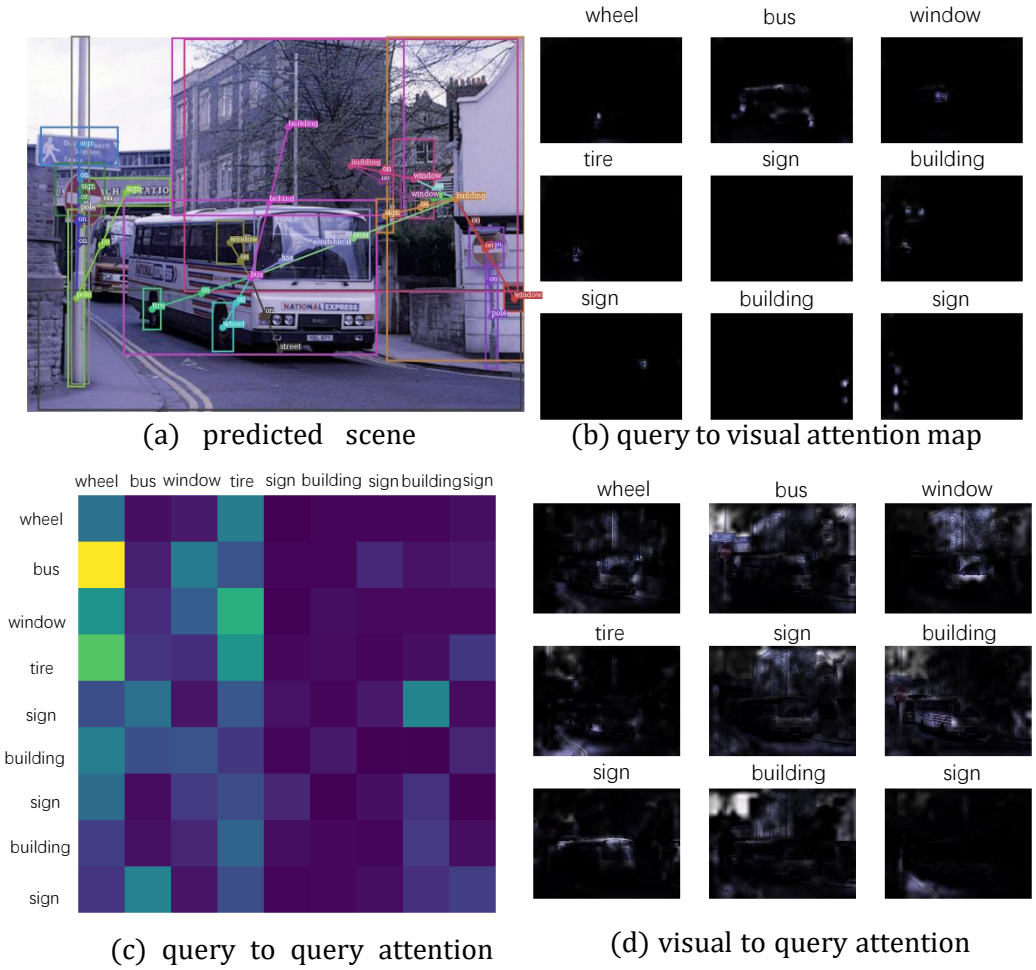


图 4-1 模型在 Visual Genome 数据集中测试图片上的预测结果和可视化展示

4.5 结果分析

为了分析模型是否按照所预期的那样工作，我们对模型中注意力层进行了可视化展示和分析。

如图 4-1 所示，图中包含了四个部分，左上角是模型生成的场景图，图片取自于 Visual Genome 数据集的测试集。右上角是查询图分支到视觉分支的 attention，即查询图节点作为查询，在特征图聚集视觉特征，从这张图中，可以看到查询节点都聚焦对应的物体上，例如最终的被分类成“bus”的节点，它在特征图中聚焦在公交车区域，物体节点的特征最终将被用来分类和回归，注意力聚焦在物体的

边缘，这和 DETR 的 decoder 的可视化结果类似。图中左下角是场景图查询的自注意力层中的可视化注意力权重图，正如在第 3.3 节所述的那样，在查询图的自注意力层中，特征在图节点之间传递，消息应该在共现频率高且相关的物体对之间传递，而可视化的结果和我们所预期的相符，注意到图中“bus”和“wheel”、“building”和“sign”具有较大的注意力权重。右下角是对视觉分支对图查询分支的交叉注意力机制，即视觉特征作为查询，场景图查询向量作为键值，这张图说明表示单独含义的物体查询向量经过这层交叉注意力之后将会散布到特征图何处，从可视化结果可以发现，查询图特征被散布到整张图片中，并不会像右上角那幅图那样，聚焦在特定的物体区域之内，这和我们所设想的不一致，我们所期望的是，具有特定语义的查询特征应该散布到视觉上对应的区域。

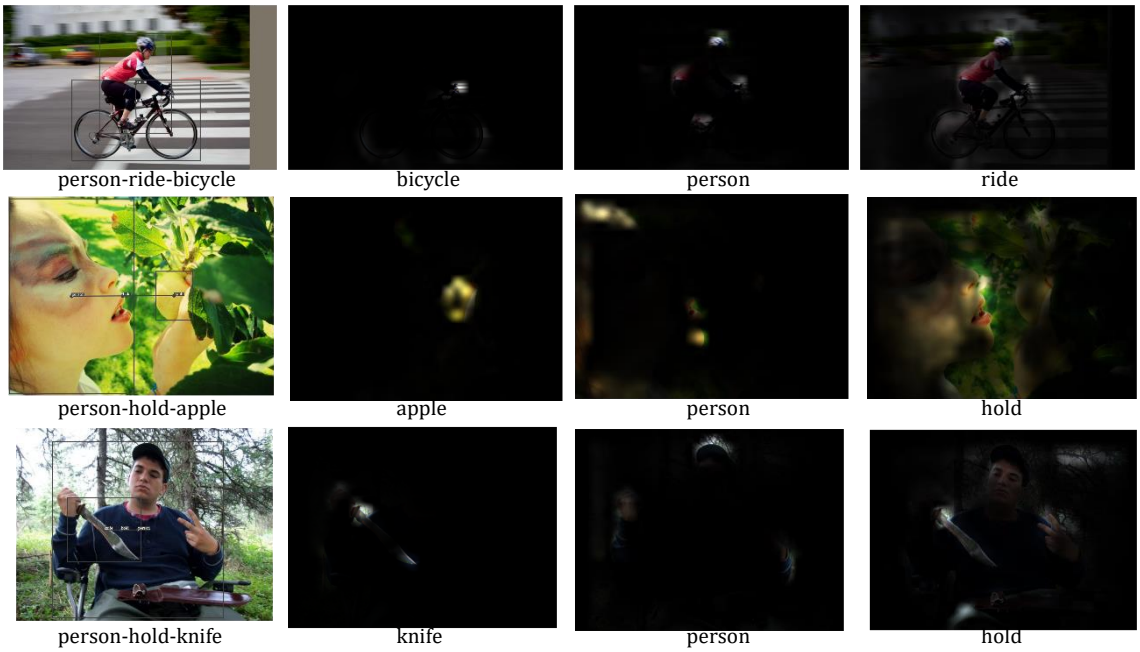
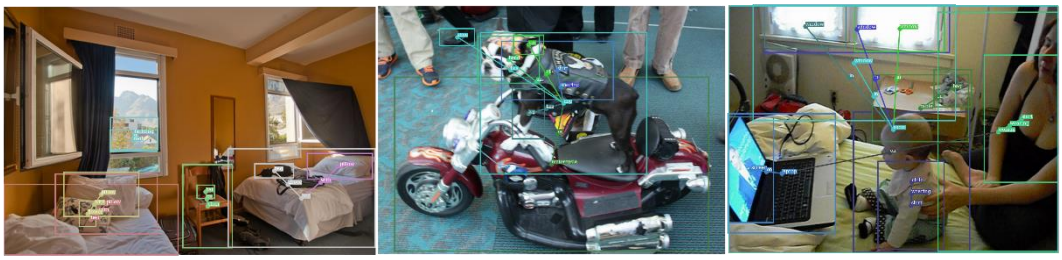


图 4-2 模型在 HICO-DET 数据集中测试图片上的预测结果和可视化展示

我们也对在 HICO-DET 数据集上的注意力矩阵进行了可视化展示，如图 4-2 所示，第一列为检测出的视觉关系，第二列为物体查询所聚焦的位置，第三列为人查询所聚焦的位置，最后一列是谓词关系所聚焦的位置。与在 Visual Genome 数据集上类似，物体查询和人查询均聚焦在物体和人的边缘以便为了更好的获得更加准确的包围框。由于我们在关系分支中加入了隐含关系查询，谓词关系所对

应的视觉区域变得不容易追溯，这里我们只是从中可视化了一张结果比较可信的注意力热图。



(a) results on Visual Genome dataset

(b) results on HICO-DET dataset

图 4-3 模型在 Visual Genome 数据集和 HICO-DET 数据集中测试图片上的预测结果

在图 4-3 中我们展示了更多模型生成的场景图实例，从结果看出，我们模型已经可以生成一些结果可信的场景图，更多的实例被放到了附录。在检查模型输出场景图的时候，我们也发现了一些不好的结果，我们的模型和绝大多数模型类似都面临着严重的偏倚问题，即倾向于去预测成那些出现频率高的关系，我们模型中没有对偏倚问题做出特殊处理，得到这样的结果也不意外，而且我们的模型虽然能够对位置相关的关系很清楚的分类，但是对于那些语义关系很困难，尤其是在 HICO-DET 数据集上，很难去把控人和物之间的语义关系，这也导致了我们的模型性能明显不如当前的主流模型。

第 5 章 总结与展望

5.1 本文工作总结

由于两阶段算法自身存在一定的局限性，本文针对如何搭建一个端到端的模型展开探索。由于 Transformer 自身非常适合处理图类型的数据，而如今存在将 Transformer 应用到其他任务的方法，所以本文选取 Transformer 作为自己模型的主干，最终，本文提出了一种针对场景图生成任务的端到端模型，大量的实验表明我们的模型可行，并且在数据集上取得了不错的结果。本文的主要工作可以总结如下：

1）分析了目前两阶段算法和单阶段算法的局限性。目前两阶段算法由于其搭建在目前业界成熟的目标检测模型之上，在训练过程中，分成多阶段训练或者需要加载目标检测的预训练模型，在预测过程，需要使用启发式算法来过滤候选框，而且具有 RPN 的方法对于视觉关系的特征不能很好的把控。单阶段算法模型漂亮且效率高，但是它依赖人工构造的结构和复杂的后处理操作。本文以此为动机，探索如何搭建一个端到端的模型来解决这些问题。

2）使用 Transformer 构建出一个端到端的模型。本文利用 Transformer 中的注意力机制在多张图中融合特征，即在特征图中的二维网格、场景图查询构建的完全图、由外界知识所构建的知识图谱这三张图中进行推理。采用在不同图空间之间投影和推理的方式来不断增强场景图查询分支中的特征。最终的场景图查询特征被去回归物体的位置和分类物体、关系的类别，最终构建出场景图，这个过程中没有引入任何人为的启发式干预，是一种真正端到端的模型。

3）构建了完全图查询结构并对此做出优化。本文采用一种非常简洁的场景图表示方法，构建完全图，列举出所有可能的物体对，而且这种完全图结果对于端对端模型是必要的。由于图片的关系十分稀疏，而查询中的关系过于密集，由此造成的不平衡会引来过多的无效计算，针对此问题，本文提出了使用交叉注意力机制来从密集关系中自适应地选择可能正确

的潜在关系，由此在保留原有的完全图结构的前提下有效缩减了计算复杂度。

4）将外部知识引入到模型中。为了进一步缓解模型对训练数据规模的需求压力和进一步提高模型的性能，本文采用了大多数方法那样将外部知识引入到模型中，利用由外部知识所构建出知识图谱仿照人类大脑来构建语义空间，并且在推理的过程中，使用交叉注意力机制来将带有视觉特征的场景图投影到该语义空间中，进行各种概念的推导。最后被语义特征增强的场景图查询特征又被投影到原特征图中来微调特征。而后的实验表明，融入外部之后可以提高模型的拟合数据的能力。

5）为模型设计了两个新的损失函数。为了对已标注的数据进行强有力的监督，并且对未标记的数据进行强有力的召回，本文对关系分支设计了分层损失，通过设置不同的损失权重使模型能够正确区分已标注并挖掘那些潜在的正确的未标注关系。为了使加入外部知识的分支能够正确把握知识，本文针对外部知识新增了一个损失，将最终的场景图特征投影到不同的语义空间中，使得在新的语义空间中，这些概念能够相互关联。

6）进行了大量的实验来验证模型。本文在两个主流的数据集：Visual Genome 数据集和 HICO-DET 数据集展开实验，并和目前的方法相比较，结果显示模型可行，但是和最近刚刚提出的模型还存在差距。与此同时，还在数据集上做出消融实验，在 Visual Genome 数据集上的实验结果表明，我们的模型中在不同图之间投影的结构相比 DETR 中基于编码解码器的结构更适合处理场景图，而且加入额外的损失之后可以提供强有力的监督，引入外部知识之后可以提高模型的表达力。在 HICO-DET 数据集表明，限制我们模型性能的因素不在于诸如 Transformer 子层的层数和查询图的规模，而在于损失的设置和模型的结构。

5.2 未来工作展望

本文针对场景图生成任务提出了一种端到端的模型，但是由于其受数据集分布不均而带来的偏倚问题以及不能很好地处理带有语义的关系，模型的一些细微之处还存在改进的空间。具体来说，

1）完全图查询结构的优化。本文所提出的模型的输入一张密集连接的完全图，并使用隐含关系查询来将关系分支的复杂度由$O(n^2)$降到$O(n)$，而且在本文所提出的模型中，为了不引入其他额外的后处理，这种完全图又是必要的，因为在完全图中关系的位置决定它和什么物体相连接。那么进一步考虑，如果完全图的引入仅是为了自动关联和它邻接的两个节点，那么为什么需要在输入就创建一个完全图结构呢？这显然是下一步需要改进的地方，一种改进的方案是，从模型输入到模型整个结构均使用隐含关系，只在输出将隐含关系投影到完全图中，这样保证了完全图可以隐含关联物体和节点，并且在模型的主干结构没有引入过多的冗余关系。

2）损失的设置。由于数据集分布不均，模型会存在很明显的偏倚问题，一种解决办法是使用重新加权的方式，对出现次数较少的关系设置较大的损失权重，这样当网络把这些网络预测错误之后会带来较大的惩罚，使之按照正确的方向调整参数。另外，在实验中我们也发现知识投影损失会影响原先用于回归和分类的特征，但又为了保证在引入外部知识之后，网络能够正确利用这些知识。又需要使用额外的损失辅助网络按照正确的方向调整参数，如何设置知识损失需要之后进一步探索。

3）外部知识的引入。本文利用外部知识来构建知识图谱，并且将视觉特征投影到这种带有语义的知识图谱上，在这张知识图谱上做完推理之后，反映射到视觉特征空间中，但是这种结构从结果上看虽然提高了模型在训练集上的性能，但是是否真的是数据过拟合导致在验证集上性能下降的需要进一步探索。

从这个领域的发展上看，未来的研究点可能存在于场景图结构的重新优化、数据集的构建、自监督学习的引入、外部知识的自动构建，具体而言，

1）场景图不能表示层级关系。图相比树结构是一种广泛意义上的结构表示，然而图中节点的连接没有限制，这导致了像层级关系不能很好的被图结构所表示。另一方面，数据集中大量存在层级关系，例如“man-has-head”、“man-of-hand”、“man-wears-shorts”等，“head”只可能和它从属的属主即“man”存在有意义的关系，这样的从属关系的存在使得场景图变得非常稀疏。如何表示这样的从属关系，场景图并不能胜任。我们提出一个

可行的解决方案，结合图和树的特点，构建树图结构，在树图结构中，节点不再仅表示图片中单个实体这个概念，而代表一个按照从属关系所构建出的树，即在上面的例子中，“man”、“head”、“hand”和“shorts”被划分到场景图的统一节点，而在这个节点表示以“man”为父节点，以“hand”、“head”和“shorts”为子节点的树。一种方式把场景图包含从属关系的节点组织成树结构并合并到单个节点，将会极大的降低场景图的稀疏度，会便于后续的处理。

2）构建一个新的数据集。目前 Visual Genome 数据中的关系是从图片标题中抽取出来的，按照这种方式构建的场景图适合帮助图片标题自动生成，但是不足于满足视觉问答等其他任务的需求。数据集存在诸如“near”、“behind”大量的关于空间位置关系，语义关系很少，而且关系分布不均，“on”关系占很大比例，这将导致模型在训练过程会偏向“on”。需要一个新的数据集来克服上述局限。

3）引入自监督学习。自监督学习已经在自然语言处理被应用，提高了下游任务的性能。类似地，自监督学习能否引入到场景图使网络只输入图片，自动从图片中挖掘结构信息呢？对比学习近两年来备受关注，它通过对比图片的不同之处，来挖掘图片的具有区分度的特征。将对比学习的思想，使用对比学习来分析图片中所普遍出现的模式，比如“player”和“racket”总是同时出现，那么这两者就可能存在关系，而且自监督学习的引入和缓解模型对数据需求的压力。

4）外部知识的自动构建。本文所采用的外部知识有一部分是人为构建的，这部分知识只能用于该数据集，更换数据集之后需要重新构建，这不符合现实应用的场景。目前存在自动从数据中学习外部知识的方法，通过设置损失训练网络，网络可以把知识储存在网络的参数中，这种方法很有吸引力，未来的工作可能会向这个方向发展。

第 6 章 结论

本文提出了一种针对场景图生成任务的端到端的模型，大量的实验表明我们的模型可行，并且在数据集上取得了不错的结果，但是和目前主流方法的性能缓存在差距，本文提出的模型存在很强的偏倚问题以及不能很好地处理带有语义含义的关系，模型需要继续进行后续的迭代优化。

致　　谢

本文的完成并未一己之力，而是站在巨人的肩膀上，感谢前人在与本文相关的领域所作出的贡献，他们的工作对本文的工作有很大的启发。

其次感谢我的导师对本文工作的支持和指导。她在我迷茫的时候为我点亮前进的指路灯，使我能够及时回归到正确的道路上。在学术上，对我的每一次迸发出的想法，虽然想法有些许幼稚，但是她都在认真聆听并指出我考虑不周到的地方。在生活上，也给予了我必要的资助。

另外，我十分感谢上海科技大学对本文工作所提供的学术氛围和实验环境。上科大是一所年轻且具有活力的大学，在上科大度过的这几个月，浓厚的学术氛围极大感染了我。本文所需的 GPU 资源，也由上科大所提供，高性能 GPU 的支持使得我能够在短时间内进行大量的实验。

最后我要着重感谢我的母校对我的培养，这四年所学的专业知识为我之后从事相关工作和研究打下坚实基础，四年和师生的接触也对我人格有正向塑造。

参考文献

[1] J. Johnson, R. Krishna, Michael Stark, L. Li, D. Shamma, Michael S. Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015.

[2] H. Xu, Chenhan Jiang, Xiaodan Liang, Liang Lin, and Zhenguo Li. Reasoning rcnn: Unifying adaptive global reasoning into large-scale object detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6412–6421, 2019.

[3] Marcel Hildebrandt, H. Li, Rajat Koner, Volker Tresp, and Stephan Günnemann. Scene graph reasoning for visual question answering. *ArXiv*, abs/2007.01072, 2020.

[4] Z. Fei. Better understanding hierarchical visual relationship for image caption. *ArXiv*, abs/1912.01881, 2019.

[5] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018.

[6] Yibing Zhan, J. Yu, T. Yu, and D. Tao. On exploring undetermined relationships for visual relationship detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5123–5132, 2019.

[7] Gengcong Yang, J. Zhang, Yanxin Zhang, Baoyuan Wu, and Yujiu Yang. Probabilistic modeling of semantic ambiguity for scene graph generation. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[8] Yikang Li, Wanli Ouyang, Zhou Bolei, Shi Jianping, Zhang Chao, and Xiaogang Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. In *ECCV*, 2018.

[9] Danfei Xu, Yuke Zhu, C. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. *2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 3097–3106, 2017.

[10] J. Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. *ArXiv*, abs/1808.00191, 2018.

[11] Mengshi Qi, Weijian Li, Zhengyuan Yang, Y. Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3952–3961, 2019.

[12] Long Chen, Hanwang Zhang, Jun Xiao, X. He, Shiliang Pu, and S. Chang. Counterfactual critic multi-agent training for scene graph generation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4612–4622, 2019.

[13] Xin Lin, Changxing Ding, Jinquan Zeng, and D. Tao. Gps-net: Graph property sensing network for scene graph generation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3743–3752, 2020.

[14] Tianshui Chen, Weihao Yu, R. Chen, and L. Lin. Knowledge-embedded routing network for scene graph generation. *2019 IEEE/CVF Conference on Computer*

Vision and Pattern Recognition (CVPR), pages 6156–6164, 2019.

[15] Hengyue Liu, Ning Yan, Masood S. Mortazavi, and B. Bhanu. Fully convolutional scene graph generation. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[16] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310, 2017.

[17] Rajat Koner, Poulami Sinhamahapatra, and Volker Tresp. Relation transformer network. *ArXiv*, abs/2004.06193, 2020.

[18] Meng-Jiun Chiou, R. Zimmermann, and Jiashi Feng. Visual relationship detection with visual-linguistic knowledge from multimodal representations. *IEEE Access*, 9:50441–50451, 2021.

[19] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *ECCV*, 2020.

[20] Cewu Lu, R. Krishna, Michael S. Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016.

[21] Georgia Gkioxari, Ross B. Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8359–8367, 2018.

[22] Ankan Bansal, Sai Saketh Rambhatla, Abhinav Shrivastava, and R. Chellappa. Detecting human-object interactions via functional generalization. In *AAAI*, 2020.

[23] Zhi Hou, Xiaojiang Peng, Yu Qiao, and D. Tao. Visual compositional learning for human-object interaction detection. In *ECCV*, 2020.

[24] Chen Gao, Jiarui Xu, Yuliang Zou, and Jia-Bin Huang. Drg: Dual relation graph for human-object interaction detection. In *Proc. European Conference on Computer Vision (ECCV)*, 2020.

[25] Yue Liao, Si Liu, F. Wang, Yanjie Chen, and Jiashi Feng. PPDM: Parallel point detection and matching for real-time human-object interaction detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 479–487, 2020.

[26] Tiancai Wang, Tong Yang, Martin Danelljan, F. Khan, X. Zhang, and J. Sun. Learning human-object interaction detection using interaction points. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4115–4124, 2020.

[27] Xubin Zhong, Xian Qu, Changxing Ding, and Dacheng Tao. Glance and gaze: Inferring action-aware points for one-stage human-object interaction detection. In *CVPR*, 2021.

[28] Cheng Zou, Bohan Wang, Yue Hu, Junqi Liu, Qian Wu, Yu Zhao, Boxun Li, Chenguang Zhang, Chi Zhang, Yichen Wei, and Jian Sun. End-to-end human object interaction detection with hoi transformer. In *CVPR*, 2021.

[29] Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. QPIC: Query-based pairwise human-object interaction detection with image-wide contextual information. In *CVPR*, 2021.

[30] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J. Kim.

HOTR: End-to-end human-object interaction detection with transformers. In *CVPR*, 2021.

[31] Meida Chen, Yue Liao, Si Liu, Zhiyuan Chen, F. Wang, and Chen Qian. Reformulating hoi detection as adaptive set prediction. In *CVPR*, 2021.

[32] Y. Li and A. Gupta. Beyond grids: Learning graph representations for visual recognition. In *NeurIPS*, 2018.

[33] Xiaodan Liang, Zhiting Hu, Hao Zhang, L. Lin, and E. Xing. Symbolic graph reasoning meets convolutions. In *NeurIPS*, 2018.

[34] Y. Chen, Marcus Rohrbach, Zhicheng Yan, S. Yan, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 433–442, 2019.

[35] Songyang Zhang, Shipeng Yan, and Xuming He. LatentGNN: Learning efficient non-local relations for visual recognition. In ICML, 2019.

[36] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exploring context and visual pattern of relationship for scene graph generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8188–8197, 2019.

[37] Tianyi Wu, Yu Lu, Yongfeng Zhu, Chuang Zhang, Miaonan Wu, Zhanyu Ma, and G. Guo. Ginet: Graph interaction network for scene parsing. In *ECCV*, 2020.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[39] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and M. Zhou. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, 2020.

[40] Weijie Su, X. Zhu, Y. Cao, B. Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pretraining of generic visual-linguistic representations. *ArXiv*, abs/1908.08530, 2020.

[41] Yen-Chun Chen, Linjie Li, Licheng Yu, A. E. Kholy, Faisal Ahmed, Zhe Gan, Y. Cheng, and Jing jing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

[42] A. Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, M. Dehghani, Matthias Minderer, G. Heigold, S. Gelly, Jakob Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020.

[43] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[44] S. Zheng, J. Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, P. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *ArXiv*, abs/2012.15840, 2020.

[45] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 4055–4064. PMLR*, 10–15 Jul 2018.

[46] R. Krishna, Yuke Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, Stephanie Chen, Yannis Kalantidis, L. Li, D. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016.

[47] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2018.

[48] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017.

[49] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685, 2018.

[50] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018.

[51] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6619–6628, 2019.

[52] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2019.

[53] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3746–3753, 2020.

[54] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[55] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. *arXiv preprint arXiv*:2001.02314, 2020.

[56] Ji Zhang, Kevin J. Shih, A. Elgammal, A. Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11527–11535, 2019.

[57] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5532–5540, 2017.

[58] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang

Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 335–351, 2018.

[59] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *Advances in neural information processing systems*, pages 2171–2180, 2017.

[60] Apoorva Dornadula, Austin Narcomey, Ranjay Krishna, Michael Bernstein, and FeiFei Li. Visual relationships as functions: Enabling few-shot scene graph prediction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[61] Tao He, L. Gao, Jingkuan Song, Jianfei Cai, and Yuan-Fang Li. Learning from the scene and borrowing from the rich: Tackling the long tail in scene graph generation. In *IJCAI*, 2020.

[62] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1261–1270, 2017.

[63] Hai Wang, W. Zheng, and Yingbiao Ling. Contextual heterogeneous graph network for human-object interaction detection. In *ECCV*, 2020.

[64] C. Gao, Jiarui Xu, Yuliang Zou, and J. Huang. DRG: Dual relation graph for human-object interaction detection. In *ECCV*, 2020.

[65] Zhi Hou, Yu Baosheng, Yu Qiao, Xiaojiang Peng, and Dacheng Tao. Affordance transfer learning for human-object interaction detection. In *CVPR*, 2021.

附录 1 英文原文

An End-to-end Model for Scene Graph Generation using Transformer

Chaofan Huo
SHANDONG University
1350133767@qq.com

Abstract

Scene graph generation aims at extracting structural representation from image to get holistic description for image content. In recent years, we have witnessed many robust models which are built upon RPN-based detector. However, their models usually need multi-stage training and their architectures are complex with stacked multi-stage modules. We pursue a more compact and concise model. At the same time, there already exists successful end-to-end models for object detection and human-object interaction. In this paper, we will follow their works to explore whether it is possible to build an end-to-end model for scene graph generation. We have successfully built an end-to-end model for scene graph generation using Transformer and we call this model as SGGTR. It achieves 28.9 recall@100 in Visual Genome dataset and 21.94 mAp score in HICO-DET dataset which is approaching the state-of-the-art models' performance. We also give thorough analysis why our model's performance is not as good as we expect. The code will be available at https://github.com/MoChen-bop/SGGTR.

1 Introduction

Humans perceive world around us in a holistic way by capturing different kinds of relationships between objects. Rather than just recognizing objects in an image, it's also important to analyze how they are organized to depict the whole content of an image jointly. Scene graph first proposed in [1] provides a logical representation to describe the high-level semantic content of an image, in which graph nodes represent objects and relationships between these nodes are represented as graph edges. Scene graph generation aims to extract scene graph from image to get a holistic description of image's content. Recent works unveil that scene graph can benefit downstream tasks, such as image retrieval [1], object detection [2], visual question answering [3], image captioning[4] and so on. Numerous works have been proposed to tackle this task. However, due to highly inner variance of visual pattern, long-tails distribution [5], incomplete annotation of dataset [6], semantic ambiguity [7], it is still a challenging task.

Existing methods are mostly built upon the state-of-the-art object detector. They first use RPN-based detector to find all possible objects' position, then apply their relationship classification branch to get relationship's category and compose these detected objects and relationships into scene graph. However, these two-stage methods have three obvious limitations. Firstly, their methods form a multi-stage pipeline. During training, they need to

train an object detector or load a pretrained model before training relation detection branch. During inference, object detector will produce thousands of proposals and even much more pair-wise relationships. It needs to use heuristic method to score and filter these objects and relationships. Secondly, they split up object detection and relationship detection. The performance of relationship detection branch relies heavily on object detection branch. If an important object which has many relationships with other objects was not detected, then there is no chance to detect all relationships connected to this object. Thirdly, visual features for objects and relationships are extracted from feature map by RoI-align operation just like most RPN-based methods did. This will cause a problem when difference kinds of relationship's region overlap just shown in [8]. Due to the lack of adaptive feature selection, simply cropping feature from corresponding region in feature map will introduce noise which will cause ambiguity for relationship classification. In this work, we attempt to tackle these three problems above. For first problem, we stack CNN and Transformer to form a simple yet powerful unified network in an end-to-end manner. As shown in figure 1, we feed graph query and CNN features into Transformer to aggregate graph features and the output of Transformer is scene graph. There is no complex postprocess needed and the output is what we want. For second problem, we combine object detection branch and relationship branch in parallel to fuse context between objects and relationships. In self-attention layer of Transformer, object-to-object, object-to-relationship, relationship-to-object, relationship-to-relationship messages can be transmitted simultaneously which forms a much more flexible reasoning routine. For the last problem, we use cross-attention to adaptively select more fine-grained features from feature map. Our contributions can be summed as following,

- We proposed a truly end-to-end network for scene graph generation using Transformer.
- To reduce the scale of pair-wise relationships, we propose to use cross-attention to reduce the complexity of relationship from $O(n^2)$ to $O(n)$.
- Our method is approaching state-of-the-art performance.

2 Related Works

Scene graph generation Existing methods [9, 10, 11, 12, 13, 14] for scene graph generation are almost built upon RPN-based detector. They first use RPN to generate possible object's bounding boxes and extract visual features from each bounding box, then use these visual features to build graph and apply iterative massage passing to refine graph nodes' features on this graph, in final prediction stage relationships are classified by drawing clues from visual features, linguistic features and spatial features. They are all two-stage method that generates object's bounding box in first stage, then predict relationship between objects in second stage. [15] inspired by previous human pose estimation methods proposes to detect objects and relationships simultaneously by treating scene graph detection as keypoint estimation. Their method significantly reduces inference time compared with two-stage method. But there still exists human effort in their pipeline. Our method is different from these methods. We build a truly end-to-end network and eliminate human-designed component, such as RAFs [16] in [15],

and complex postprocess. Our network is quite simple and the output of network is what we want.

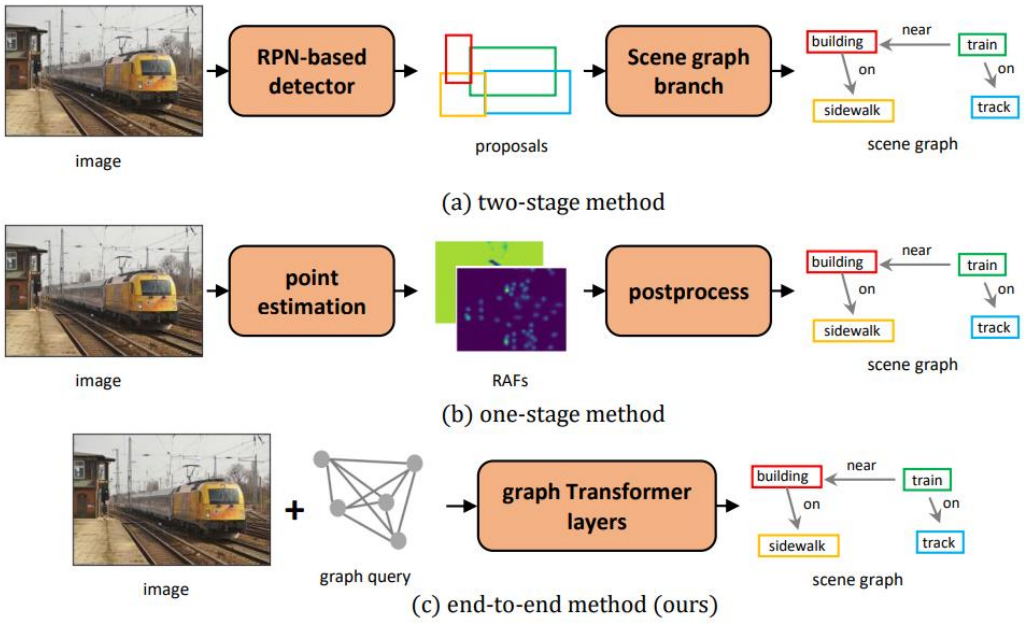


Figure 1: In two-stage method (figure 1(a)), model is designed in a multi-stage pipeline. They use RPN-based detector to find object's possible position and use scene graph branch to prediction the components of scene graph. In one-stage method (figure 1(b)), they treat scene graph generation as keypoints estimation. Their model is both simple and efficient, but they need human-designed structure, such as RAFs, and need complex postprocess in inference stage. Our model (figure 1(c)) is a truly end-to-end model which takes image and query as input and outputs scene graph, there no human-designed components or complex postprocess.

Human-object interaction Human-object interaction is a relevant task to scene graph generation. What's different is that human-object interaction targets at generating human-centric graph and relationships in an image are not as dense as scene graph generation. There are many similarities in methods between these two tasks. Methods for human-object interaction can also be divided into three branch: two-stage [21, 22, 23, 24], one-stage [25, 26, 27], end-to-end [28, 29]. There actually exists methods [28, 29, 30, 31] that use Transformer to build end-to-end model. But their methods are targeting at detect human-verb-object triplets from images. Our method can cover more general cases in which the input and output is a graph.

Graph reasoning beyond convolution To capture long-range dependencies and surrounding context, several works [32, 33, 34, 35] attempt to integrate graph models into their networks. They adopt project-reasoning-reproject steps to gather semantic information from feature map and build graph to swap features among these nodes and finally reproject graph onto feature map to refine features. Our method also incorporates these three steps into our modules by using cross-attention of Transformer. Different from [36] that using RPN region to restrict spatial position, we use cross-attention to adaptively select features and reproject features.

Co-reasoning with external knowledge There are two kinds of external knowledge. The first one is word embedding vector which is learned from a large scale language corpus which reflects the semantic similarity between words. The second one is commonsense knowledge constructed by human labor, such as man and woman are both subtypes of person [19]. Inspired by previous works [2, 37], we also incorporate these external knowledges into our model. We follow [19] to use both word semantic embedding and commonsense knowledge graph.

Transformer's application in CV Transformer [38] has achieved great performance in machine translation. In recent two years, overwhelming works have applied Transformer to various tasks in CV, such as multi-modality learning [39, 40, 41], image classification [42], object detection [43], semantic segmentation [44] and so on. Although there are several works [13, 18, 17] which apply Transformer for scene graph generation or visual relationship detection. However, they use transformer to fuse features between graph nodes and they are still two-stage RPN-based method. Different from their work we follow [43] to form a truly end-to-end model in which detection branch is based on encoder-decoder structure rather than RPN-based.

3 Approach

As shown in figure 2, there are three types input of our model, (1) visual feature sequence, (2) graph queries, (3) external knowledge. We feed them into stacked graph Transformer layers to gather information from multi-source input using attention mechanism. Our main branch of graph Transformer layer is designed in a cascaded manner to generate scene graph step by step. After graph queries aggregate enough features, it will be use to predict object categories, object positions and relationship categories of scene graph.

3.1 Multi-source input

Visual branch Given an image I with shape $\mathbb{R}^{w\times h\times 3}$, we use CNN to use extract feature map $F \in \mathbb{R}^{\frac{w}{s}\times\frac{h}{s}\times c}$ from it, where s is the stride of CNN, c is the feature map's channel. Due to Transformer's input can only receive 1D feature sequence. We flatten 2D feature map into 1D feature sequence $\{\mathbf{f}_i\}_{i=1}^{n}$ with length n like most methods did. To preserve 2D spatial position information we follow [43] to use sine-like position embedding [45].

External knowledge branch We follow [19] to use word embedding selected from GloVE according to objects' and relationships' category in dataset and prior relationships among objects gathered from ConceptNet to build our external knowledge graph. Formally, we donate external knowledge graph as $\mathcal{G}^S = \{\mathcal{V}^S, \mathcal{E}^S\}$. It has two components: graph nodes $\mathcal{V}^S$ and relationships $\mathcal{E}^S$ between these nodes. The graph nodes cover all object categories and relationship categories in dataset. Take VG dataset for example, we will have 150 object nodes plus one no-obj node and 50 relationship nodes plus one on-rel node. Each node has its own word embedding as semantic features. Relationships between these graph nodes indicate how two entity related. We recommend readers go to [19] to check more details.

Graph query branch Graph query $\mathcal{G}^Q$ is a graph in which graph nodes has no specific meanings and what is have is the topological structure. We called this graph as graph prototype for scene graph. This graph can be viewed as a memory graph which preserves all kinds of scene graph for different images, which is similar like our human's brain with which even we close our eyes, our mind can still float different kinds scenes. We fix 100 object queries and connect all object pairs exhaustedly to form a complete graph. Here the pairwise relationship is necessary for a truly end-to-end model. We didn't adopt more complex relationship queries like [30] because it will introduce more complex postprocess in inference and complex loss calculation in training stage. Like [43], we initialize all graph queries as zero vectors and append each queries with a learned position embedding to preserve topological graph structure and distinguish different graph query nodes. For each object query q_i^o the position embedding is learned and we donate it as p_i^o. For relationship between object query q_i^o and object query q_j^o, the position embedding is calculated as following,

$$p_{ij}^r = \mathrm{FC}\left(\mathrm{Concat}\left(p_i^o, p_j^o\right)\right)$$

We concatenate corresponding object queries and use fully connected layer to reduce dimension to keep it same with the objects'.

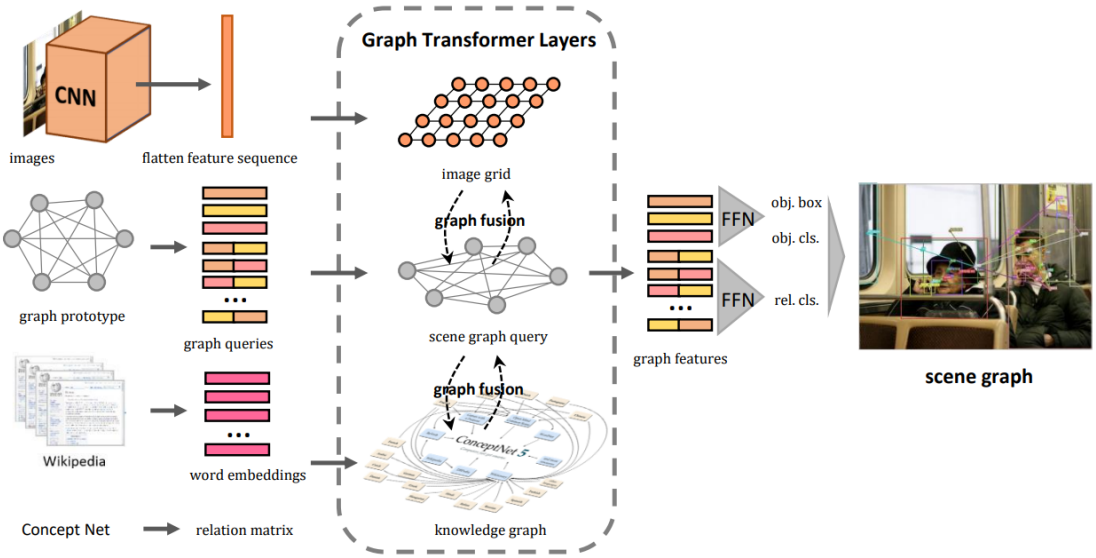


Figure 2: This figure shows main pipeline of our method. Our model has three type inputs: (1) visual branch, (2) graph query branch and (3) external knowledge branch. These multi-source inputs which are embedded as 1D feature sequences are feed into graph Transformer layers to adaptively fuse features between image grid graph, scene graph and commonsense knowledge graph. In stacked graph Transformer layers, graph query will gather features from the other branch. After several iterations, they will be fed into FFN to regress objects' box and classify objects' class and relationships' class.

3.2 Graph Transformer Module

The graph Transformer module is composed by stacked graph Transformer layers. Its input contains visual feature sequences $\{\mathbf{f}_i\}_{i=1}^{n}$, graph queries $\{\mathbf{q}_i\}_{i=1}^{m}$ and semantic word embeddings $\{\mathbf{s}_i\}_{i=1}^{k}$. The output will be scene graph features. The graph Transformer module is calculated in a cascade manner as following,

$$\{\mathbf{q}_i^{l+1}\}_{i=1}^{m} = \mathcal{G}\left(\{\mathbf{f}_i^{l}\}_{i=1}^{n}, \{\mathbf{q}_i^{l}\}_{i=1}^{m}, \{\mathbf{s}_i^{l}\}_{i=1}^{k}\right), l = 0,1,2,\dots,N \quad (1)$$

where superscript indicates that each features sequence belongs to different layers, N is the number of graph Transformer layers and $\mathcal{G}$ is one layer graph Transformer layer which we will describe below.

In graph Transformer layers, we fuse information among these three different source inputs using attention mechanism. As shown in figure 3, the information flow contains four directions: (1) from visual branch to graph query branch, (2) from graph query branch to knowledge graph branch, (3) from knowledge graph branch back to graph query branch, and (4) from graph query branch back into visual branch. First, we use graph query $\{\mathbf{q}_i\}_{i=1}^{m}$ to aggregate visual information from visual feature sequence $\{\mathbf{f}_i\}_{i=1}^{n}$ using cross-attention,

$$\{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m} = \mathrm{CrossAttention}(\{\mathbf{q}_i\}_{i=1}^{m}, \{\mathbf{f}_i\}_{i=1}^{n}, \{\mathbf{f}_i\}_{i=1}^{n}) \quad (2)$$

where $\{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m}$ is the flatten node visual features for scene graph and $\mathrm{CrossAttention}(Q,K,V)$ is a multi-head attention layer with query Q, key K and value V. After visual to query cross attention process in which graph queries select features from visual feature sequences, each graph query will be given its corresponding visual meaning. Then we project the scene graph visual features onto semantic graph which is constructed using word vectors $\{\mathbf{s}_i\}_{i=1}^{k}$.

$$\{\mathbf{f}_i^{Q\to S}\}_{i=1}^{k} = \mathrm{CrossAttention}\left(\{\mathbf{s}_i\}_{i=1}^{k}, \{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m}, \{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m}\right) \quad (3)$$

where $\mathbf{f}_i^{Q\to S}$ is the node feature in semantic graph. As described in section 3.1, semantic graph is composed using word embedding as node's semantic features and prior relationships between two words as graph edge. After we project query graph's visual features onto this semantic graph, we hope that the objects or relationships shown up in image will be activated in this semantic graph just like when we watched an image, the objects in this image will form all kinds of concepts in our mind. After this perception process, we will combine with our prior knowledge to reason how these objects are organized. To simulate this reasoning process, we use self-attention layer to do graph reasoning on this semantic graph.

$$\{\mathbf{f}_i^{S\to S}\}_{i=1}^{k} = \mathrm{SelfAttention}\left(\{\mathbf{f}_i^{Q\to S}\}_{i=1}^{k}, \mathrm{weight} = M\right) \quad (4)$$

where $\mathbf{f}_i^{S\to S}$ is the feature for graph node i after it gather features from other graph nodes in self-attention layer. We use attention weight matrix M to restrict how the messages passed in this semantic graph layer. We will describe this matrix M below in details. Note that in our semantic graph, graph nodes represent concept, i.e. object words or relationship words in dataset. These graph nodes can be split up into object graph nodes with number m_o and relationship graph nodes with number m_r. As for Visual Genome dataset, we have 202 graph nodes, 151 for object graph nodes and 51 for relationship graph nodes. Each graph nodes is represented with a semantic features $\mathbf{s}_i$ which is word embedding vector as described before. Attention weight matrix $M = (M_1, M_2, \ldots, M_k)$ is a set of relationship matrix in which each relationship matrix is a binary matrix indicating whether two concepts are related. There are k relationship matrices, each matrix has different meanings such as ISA relationship, USEFOR relationship, SUBSETOF relationship etc. These matrices is composed by [19], we recommend readers go to [19] to check more details. As shown in equation 4, we apply these matrices to self-attention weights. The motivation behind this is quite simple, we hope the network can associate related concept just like when we see "racket", we mind will float all kinds of related concepted such as "tennis", "player", "hold", "short" and so on. After this process, these semantic features in semantic graph will be enhanced with visual features and related concepts' features. Then the semantic features in semantic graph is re-project onto query graph using cross-attention again.

$$\left\{\mathbf{f}_i^{S\to Q}\right\}_{i=1}^{m} = \text{CrossAttention}\left(\left\{\mathbf{f}_i^{V\to Q}\right\}_{i=1}^{m}, \left\{\mathbf{f}_i^{S\to S}\right\}_{i=1}^{k}, \left\{\mathbf{f}_i^{S\to S}\right\}_{i=1}^{k}\right) \quad (5)$$

So far, we have gathered information from other two other branches. The initial empty scene graph queries have meaningful features aggregated from visual image and external knowledge. In order to capture objects' and relationships' dependences in image. We will apply self-attention to do reasoning over query graph.

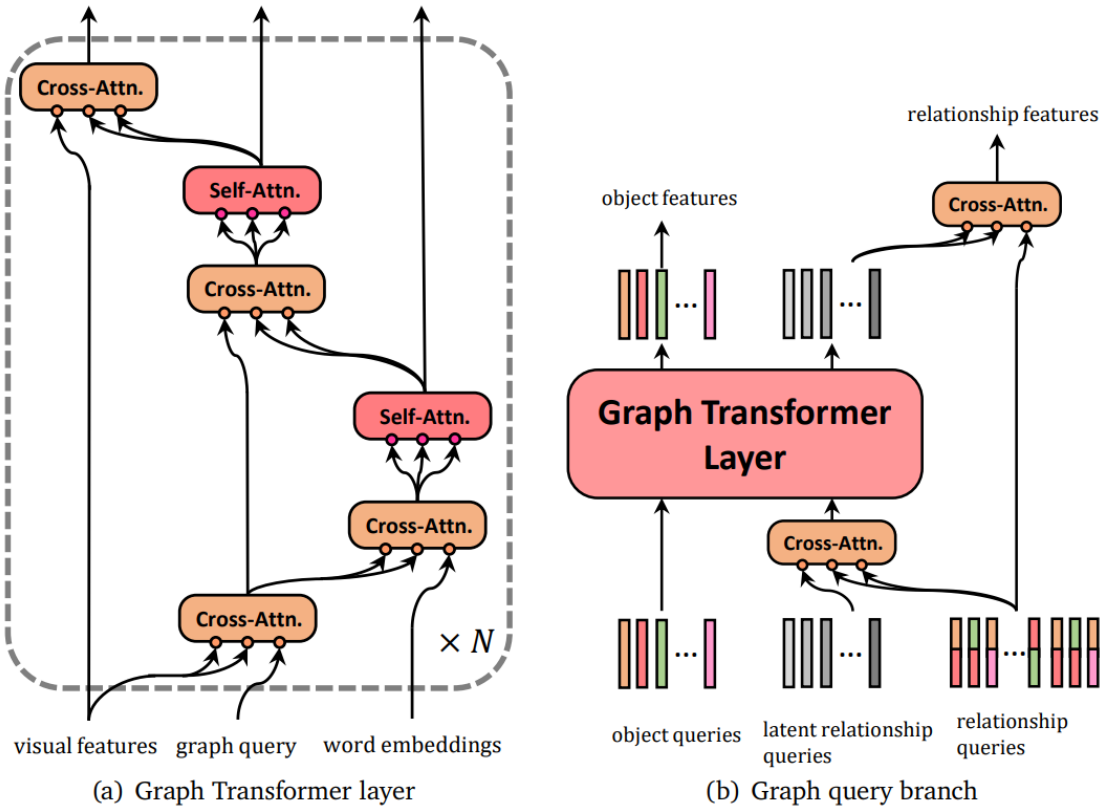
$$\left\{\mathbf{f}_i^{Q\to Q}\right\}_{i=1}^{m} = \text{SelfAttention}\left(\left\{\mathbf{f}_i^{S\to Q}\right\}_{i=1}^{m}\right) \quad (6)$$

where $\left\{\mathbf{f}_i^{Q\to Q}\right\}_{i=1}^{m}$ corresponds to $\left\{q_i^{l+1}\right\}_{i=1}^{m}$ in equation 1. In this process, objects or relationships in the same image can borrow features from each other and context can be spread among the whole image. The query features $\left\{\mathbf{f}_i^{Q\to Q}\right\}_{i=1}^{m}$ with be used to classification and regression which will be describe in section 3.4.

Finally, we reproject graph query features back onto visual features to refine original visual features.

$$\left\{\mathbf{f}_i^{Q\to V}\right\}_{i=1}^{n} = \text{CrossAttention}\left(\left\{\mathbf{f}_i\right\}_{i=1}^{n}, \left\{\mathbf{f}_i^{Q\to Q}\right\}_{i=1}^{m}, \left\{\mathbf{f}_i^{Q\to Q}\right\}_{i=1}^{m}\right) \quad (7)$$

where $\left\{\mathbf{f}_i^{Q\to V}\right\}_{i=1}^{n}$ corresponds to $\left\{\mathbf{f}_i^{l}\right\}_{i=1}^{n}$ in equation 1.



(a) Graph Transformer layer (b) Graph query branch

Figure 3: Figure 3(a) shows the data flow of graph Transformer layers. Figure 3(b) shows how we reduce the scale of relationship branch.

3.3 Reducing the Scale of Relationship

Since we exhaustedly enumerate all possible relationships in graph query, the number of relationship queries will become unacceptable as object queries increase. To tackle this problem, we utilize cross-attention to reduce the scale of relationships. As shown in figure 3(b), we define a set of latent relationship queries with fixed size. These latent relationship queries are initialized with zero vector and their position embeddings are learned during training. They are fed into cross-attention layer to select possible activated relationship queries. Any other complex operations, such as cross-attention with visual branch or semantic branch, is conducted based on these latent relationship queries. Finally, these latent relationship queries are re-project onto original relationship queries using cross-attention. In this way, we reduce the complexity of graph query branch from $O(n^2)$ into $O(n)$ while not breaking the complete graph structure.

3.4 Inference and Loss

After graph queries gather enough features from visual branch and semantic branch, they will be used for classification and regression. For object queries, we feed them into MLP layer with 3 layers to regression bounding box of object and MLP with 1 layers to classify objects'

categories,

$$\mathbf{b}_i = \mathrm{MLP}\left(\mathbf{f}_i^{Q\to Q}\right) \text{ and } c_i^o = \mathrm{MLP}\left(\mathbf{f}_i^{Q\to Q}\right) \quad (8)$$

where $b_i \in \mathbb{R}^4, c_i^o \in \mathbb{R}^{C_o+1}$, C_o is the number of object category. For relationship queries, they are used to classify categories only and there is no need to regress position for relationships, since which objects connected to relationship is implied in position embedding. So we have

$$c_{ij}^r = \mathrm{MLP}\left(\mathbf{f}_{ij}^{Q\to Q}\right) \quad (9)$$

where $c_{ij}^r \in \mathbb{R}^{C_r+1}$ is relationship between object i and object j, C_r is number of relationship category in dataset. In inference stage, we filter out all no object queries and keep relationship whose head object and tail object are classified into no-obj neither.

In training stage, we use Hungarian bipartite matching [43] to assign groundtruth to each prediction. Our loss is calculated as follow

$$\mathcal{L}_{\mathrm{total}} = \lambda_{\mathrm{L1}}\mathcal{L}_{\mathrm{L1}} + \lambda_{\mathrm{IOU}}\mathcal{L}_{\mathrm{IOU}} + \lambda_{\mathrm{CLS}}\mathcal{L}_{\mathrm{CLS}} + \lambda_{\mathrm{knowledge}}\mathcal{L}_{\mathrm{knowledge}} \quad (10)$$

where $\mathcal{L}_{L1} = \sum_i \left\|\mathbf{b}_i - \hat{\mathbf{b}}_i\right\|_1$ is l_1 loss between predicted box $\mathbf{b}_i$ and ground-truth box $\hat{\mathbf{b}}_i$, $\mathcal{L}_{\mathrm{IOU}}$ is generalized IoU loss. We use cross-entropy to calculate loss $\mathcal{L}_{\mathrm{CLS}}$ for classification. In order to alleviate bias towards no-obj and no-rel class, we set a loss weight to empty object class and empty relationship class. Compare with [43], we add another two new losses, one for loss if relationship classification, the other loss is for punishment of knowledge projection loss.

We treat the output of relationship branch is a 2D matrix with the shape $\left(n_{obj}, n_{obj}\right)$, where n_{obj} is the number of object query. In this matrix, position (i,j) is a logits vector $c_{ij}^r \in \mathbb{R}^{C_r+1}$ which represents the relationship between object query i and object query j. We create ground-truth relationship label matrix according to the assignment of Hungarian matching for object branch where position (i,j) is assigned with the relationship label between the corresponding object of object query i and corresponding object of object query j. The loss between predicted relationship logit matrix and groundtruth relationship label matrix is calculated using cross-entropy with weight of no_rel_weight for empty relationship labels. To clarify the annotated labels and undermined labels, we set two weights for relationship class. First we calculate the loss between predicted relationship logit matrix and groundtruth relationship label matrix as described above with weight no_rel_weight. Then we add auxiliary loss for annotated labels. For example, if relationship between object i and object j is annotated, then we calculated relationship loss for object i and object j with weight of aux_no_rel_weight.

As for knowledge project loss, we first project objects' features and relationships' features

into a series of semantic spaces with different semantic meaning using MLP with one layers. In each semantic layer, each object or relationship class is assigned with a semantic label defined by external knowledge graph. For example, in semantic space with meaning of "similar with", the class "wearing" are similar with "has" and "wears" which is defined by external knowledge, then "wearing" is classified with multi label "has" and "wearing". The motivation behind of this is that we hope words with similar meanings should be project into the same subspace.

Recall @ K / No-graph Constraint Recall@K		Scene Graph Detection											
		R@20/50/100			ng-R@20/50/100			mR@20/50/100			ng-mR@20/50/100		
External Knowledge	VCTree[51]	22.0	27.9	31.3				5.2	6.9	8.0			
	KERN[52]		27.1	29.8		30.9	35.8		6.4	7.3			
	GPS-NET[53]	22.3	28.9	33.2						9.8			
	MOTIFS-TDE[50][54]	12.4	16.9	20.3				5.8	8.2	9.8			
	GB-NET[19]		26.3	29.9		29.3	35.0		7.1	8.5		11.7	16.6
	RelDN[56]	21.1	28.3	32.7		30.4	36.7						
Visual Only	VTransE[57]		5.5	6.0									
	FactorizableNet[58]		13.1	16.5									
	IMP[48][50]	14.6	20.7	24.5									
	Pixels2Graphs[59]					9.7	11.3						
	Graph R-CNN[49]		11.4	13.7									
	VRF[60]		13.2	13.5									
	CISC[36]	7.7	11.4	13.9									
	HRNet[15]	16.1	21.3	25.1	16.7	23.5	29.2	2.7	3.6	4.2	3.8	5.7	7.5
	CMAT[15]	22.1	27.9	31.2	23.7	31.6	36.8						
	KERN[14]		27.1	29.8		30.9	35.8		6.4	7.3			
	LSBR[61]	23.6	28.2	31.4	26.9	31.4	36.5						
	SGGTR (Ours)	18.7	24.7	28.9	20.5	28.1	33.5	4.0	5.9	7.4	5.8	9.7	13.7

Table 1: Recall and no-graph constraint recall @K evaluation results on VG-150.

4 Experiments

4.1 Implementation details

We follow [43] to use ResNet-50 as backbone to extract feature. The output features' channel from layer 4 of ResNet50 is 2048, we downsample it to 256. We fix 100 object queries and 100 latent relationship queries with dimension of 256. We use five graph Transformer layers and in each layer we use multihead attention with 8 heads and feed forward layers' dimension is set to 2048. In loss function, we set λ_{L1} to 5, λ_{IOU} to 2 and λ_{CLS} to 1. Empty object weight is set to 0.1 and empty relationship weight is set to 0.01. We use AdamW optimizer to train this model. Learning rate is set to 5×10−5 and decays at 60 epoch. We load pretrained ResNet-50 from TORCHVISION with frozen batchnorm layers. The total training process may take 4 days.

4.2 Datasets

We use Visual Genome dataset [46] to test the performance of scene graph generation and HICO-DET dataset [47] to test the performance of visual relationship detection. The original Visual Genome dataset contains 108, 077 images. We use [48] split to train and evaluate our models, which filters out not frequent objects and relationships and left 150 object classes and 50 relationship classes. In this dataset, we report recall@20/50/100 with/without graph constraint and mean recall@20/50/100 for each class. The HICO-DET dataset contains 47, 776 images and in which 38, 118 images are used for training, 9, 658 for evaluation. We use mAp scores as evaluation metric.

4.3 Compare with the S.O.T.A. models

We collect results reported in recent four years on VG dataset and HICO-DET dataset. As shown in table 1 and table 2, our method is approaching the state-of-the-art performance but there still exists a non-negligible gap. In Visual Genome dataset, our methods achieve 18.7 recall@20 and has beaten methods which proposed before 2019, such as Graph RCNN [49], CISC [36], IMP [48, 50]. But there is still gap between recently released models. In HICO-DET methods, there is also several mAp scores dropped compared with these state-of-the-art methods.

Method		mAp		
		full	rare	non-rare
Two-stage	CHGN[63]	17.57	16.85	17.78
	DRG[64]	24.53	19.47	26.04
	VCL[23]	23.63	17.21	25.55
	ATL[64]	23.81	17.43	24.32
One-stage	PPDM[25]	21.94	13.97	24.32
	IP-Net[26]	19.56	12.79	21.58
	GGNet[27]	23.47	16.48	25.60
	AS-Net[31]	28.87	24.25	30.25
	HOTR[30]	25.10	17.34	27.42
End-to-end	HoiTransformer[38]	26.61	19.15	28.84
	QPIC[29]	29.90	23.92	31.69
	SGGTR(ours)	21.94	17.14	23.37

Table 2: Performance on HICO-DET dataset

4.4 Ablation study

To validate whether the model's performance can be improved further with additional loss. We trained SGGTR with or without knowledge loss and auxiliary loss as described in section 3.4. As shown in table 3, after we add addition loss for relationship the recall scores are dropped greatly in validation dataset. However, the model's performance in training dataset is improved. Note that when we train model on VG dataset, we didn't use any data augmentation technique. We blame this on overfitting. We will retrain these models with data augmentation and the results will be reported afterwards. And we also find that DETR-based SGGTR performs very well for object detection in training dataset. This indicate that the split encoder and decoder is critical for object detection. Compare DETR-based model and SGGTR model, SGGTR model is slightly better than DETR-based model which is encoder-decoder based. This indicates that the project-reasoning-reproject method is more suitable to process graph-based data.

Model	w/o knowledge loss	w/o auxiliary relationship loss	on eval set				on training set			
				recall@				recall@		
			mAp	20	50	100	mAp	20	50	100
DETR-Based	✗	✗	23.61	13.47	20.93	25.90	53.71	31.99	43.69	51.87
SGGTR	✗	✗	24.50	18.45	24.30	28.56	34.46	24.56	29.96	31.24
	✗	✓	24.11	11.05	16.94	22.01	36.39	39.07	48.63	54.98
	✓	✓	22.32	10.01	15.20	20.23	32.78	35.88	44.20	49.64
SGGTR with exknowledge	✗	✗	22.36	14.66	21.82	26.28				
	✓	✓	21.99	9.96	15.35	20.19	43.53	41.75	52.06	58.60

Table 3: Ablation study in Visual Genome dataset

We also compare SGGTR and DETR in HICO-DET dataset as shown in table 4. SGGTR's performance is close to DETR's. And by comparing SGGTR with 20 object queries and 50 queries, the performance has no significant difference. And this indicates that the model's architecture and number of object query are not the decisive factor. What make our method worse than these state-of-the-art models may be the loss function and design of graph query.

Model	Object query number	Relationship query number	Graph Transformer layer number	mAp		
				full	rare	non-rare
DETR	20	20	5	22.31	18.36	23.49
SGGTR	20	20	5	21.93	18.45	22.97
	20	20	6	20.51	15.68	21.96
	50	50	5	21.94	17.14	23.37

Table 4: Ablation study in HICO-DET dataset

4.5 Analysis

To analyze if our model works as we expect, we visualize multihead attention heatmap. As shown in figure 4, we obvious that graph queries in graph query to visual feature cross-attention focus on the edge of object and this behaves like decoder in [43]. However, in visual feature to graph query cross-attention, the graph query features are spread among image and don't focus on target meaningful region. This indicates that graph query features will blur original features if they are re-project onto critical regions as shown in figure 4(b) that are decisive for object regression. In graph query to graph query self-attention, we obverse message passed among correlated objects such as bus and wheel in figure 4(c). We will put more visualized results in appendix.

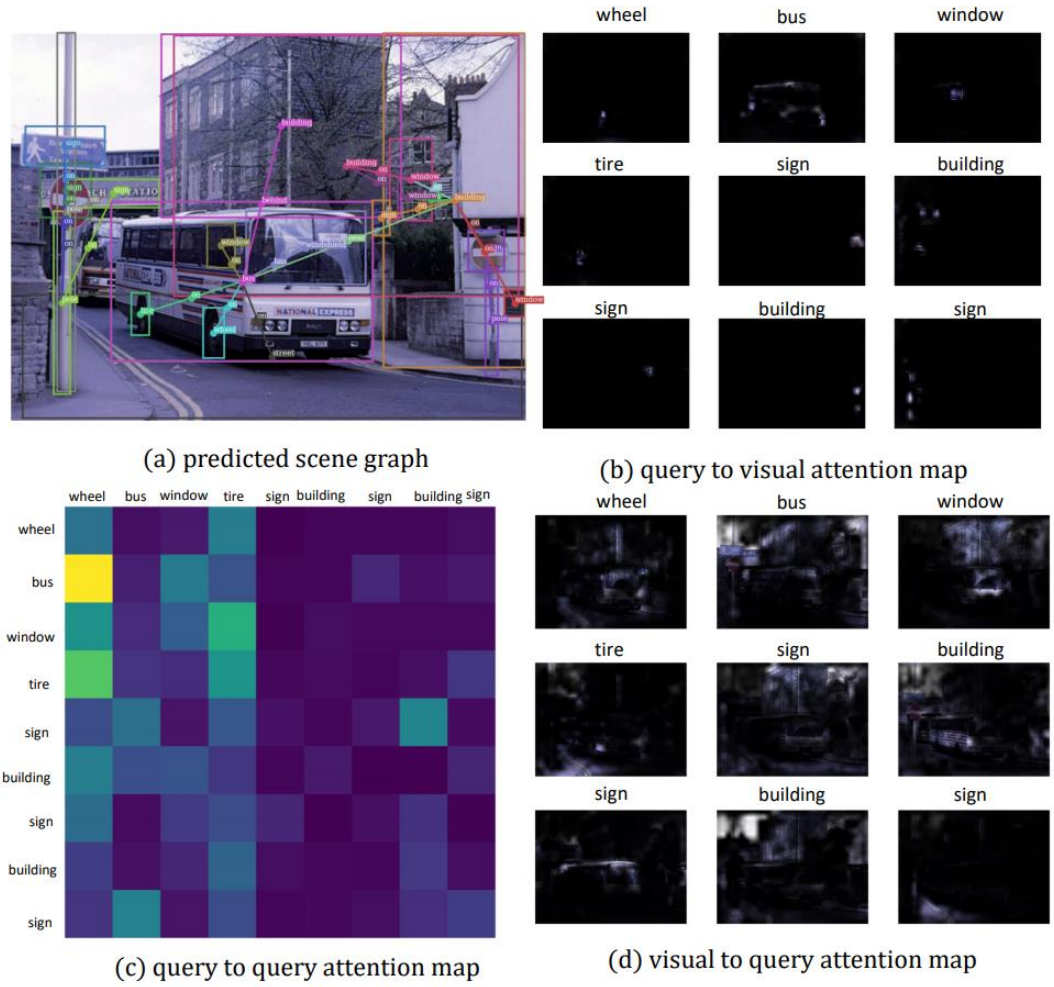


Figure 4: We visual attention heatmap in Visual Genome dataset. Figure 4(a) is scene graph generated using our model. 4(b) is attention map in cross attention between graph query and visual branch. 4(c) is attention map in self-attention among graph queries. Figure 4(d) is heatmap between visual branch and graph query branch.

We also visualize results on HICO-DET dataset, as shown in figure 5. Because we use

latent relationship queries to search visual ground, it is difficult to trace where the verb attends to. Here we visualize some reasonable heatmap for relationships.

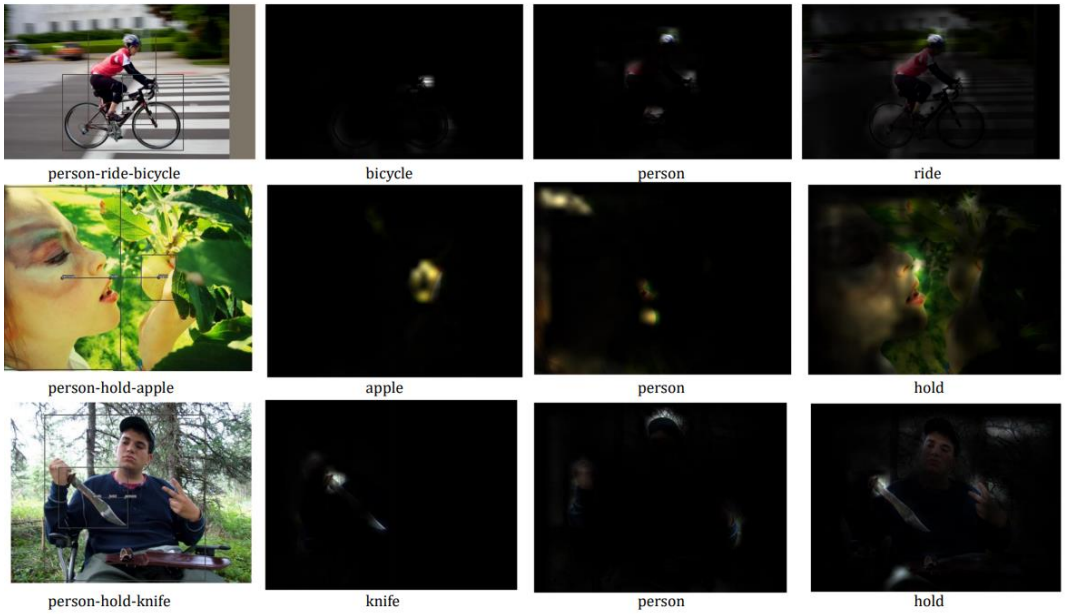


Figure 5: Visualized attention heatmap in HICO-DET dataset.

Figure 6 shows several scene graphs generated using our model. The results is reasonable. However, during we examine the scene graphs output by our model, we also found several bad results as shown in appendix. Our model has a strong bias problem as shown in [54]. Our model is inclined to predict relationship class which shows frequently in dataset. Another problem our model has is that our model struggle to capture semantic relationships such as "inspect", "direct", "load", "block" in HICO-DET dataset.

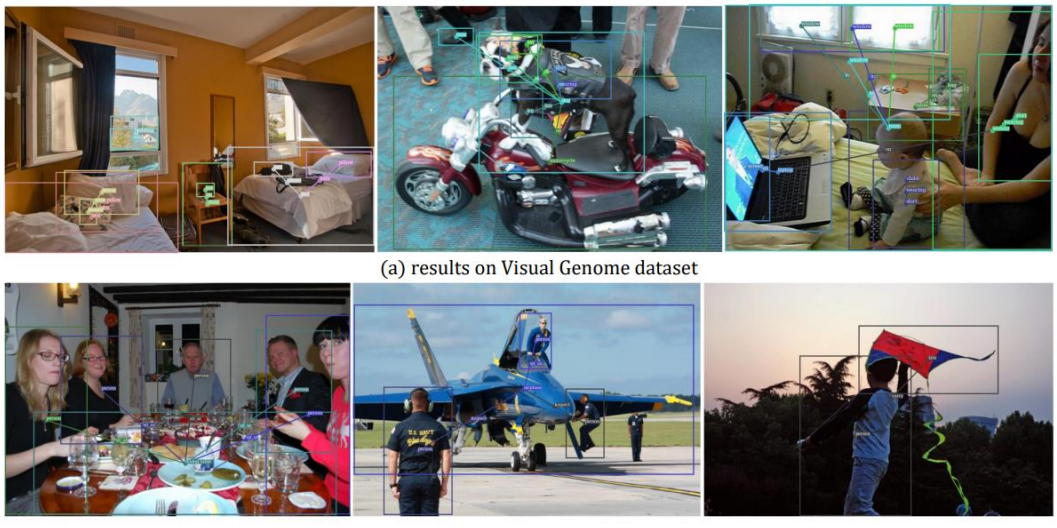


Figure 6: Generated scene graph in Visual Genome dataset and HICO-DET dataset.

附录 2 译文

一种针对场景图生成任务的端到端的模型

霍超凡
山东大学
1350133767@qq.com

摘要

场景图旨在从图片中提取结构化表示，以得到对图像内容的完整描述。近年来，许多方法搭建在基于 RPN 的物体检测器上，他们的模型通常需要多阶段训练并且他们的网络架构由多个模块堆叠形成，而我们追求一种更加简洁的模型。与此同时，已经有成功应对物体检测和人物交互检测的端对端模型。在本文中，我们将跟随他们的工作，探讨是否有可能建立针对场景图生成任务的端到端的模型。我们成功地搭建了一个基于 Transformer 的端到端的场景图生成模型，我们称之为 SGGTR。我们的模型在 Visual Genome 数据集上达到了 28.9recall@100，在 HICO-DET 数据集上达到 21.94mAp，正在接近目前主流模型的性能，最后我们还对模型性能不如预期那样好的原因进行了深入分析。代码将在站点 *https://github.com/MoChen-bop/SGGTR* 提供。

1 引言

人类通过捕捉不同物体之间的关系，以获取对周围世界的整体性认知。我们并不满足于检测出图像中的单个物体，还侧重于分析它们是如何组织起来以共同完成对图像的内容描述。场景图首次在[1]被提出，它提供了一种描述图像高级语义内容的逻辑表示方式。场景图是一种图结构模型，图中节点表示图像中的对象，节点关系表示物体之间的视觉关系。而场景图生成任务的目的是从图片中提取场景图，以得到对图像内容的整体描述。场景图生成任务是一种连接诸如物体检测底层任务到图片语义理解高层任务的桥梁，近几年的工作揭示了场景图对诸如图像检索[1]、目标检测[2]、视觉问答[3]、图片标题生成[4]等下游任务有益处。学术界针对场景图生成任务提出了很多方法，然而由于视觉层面的高度内在

差异、数据集长尾分布[5]、数据集不完全注释[6]、语义歧义[7]等问题，它至今仍然是一个具有挑战性的课题。

现存方法大多搭建在主流物体检测网络之上，首先使用基于 RPN 的检测器来寻找所有可能存在物体的位置，从候选框中截取出特征，将其送入之后的网络来判别物体之间的关系和类别，并将这些检测到的对象和关系组装成场景图。然而，这样两阶段方法存在三个明显的局限性，首先，它们是以一种多阶段的方式处理图像，在训练期间，需要首先训练一个目标检测器或加载一个预训练物体检测模型，然后再训练关系检测分支。在推理期间，目标检测器将会生成数千个候选框，这些候选框可能组装成数以万计对关系，需要启发式方法对这些关系进行评分的过滤。其次，这种两阶段的方法割裂了目标检测的视觉关系检测，关系检测很大程度上依赖于物体检测，如果一个与图片中很多物体存在关系的对象没有被检测出来，则后面的分支再也没有可能检测出与这个对象关联的所有关系。第三，如同绝大多数基于 RPN 的方法一样，两阶段方法通过 RoI 对齐的方式从特征图中提取物体和关系的视觉特征，这么做将导致一个问题，当不同类型的关系的区域高度重叠，就如同在[8]所述的那样，由于缺乏自适应选择特征的机制，这样简单地从目标区域裁剪出特征将会引入噪声，从而应发后续分类的歧义。本文试图解决上述三个问题，对于第一个问题，我们将 CNN 和 Transformer 以端对端的方式堆叠在一起，形成一个简单而有效的网络。如图 1 所示，二阶段方法需要候选框提取和目标关系检测分类这两大步骤；单阶段算法将关键点检测思想应用到这里，网络结构简洁、漂亮，但是它需要人工构建的结构和复杂的后处理操作。而我们的方法是一种端到端的网络，将图查询和 CNN 特征输入到 Transformer 中，输出即为场景图，没有任何复杂的后处理操作。对于第二个问题，我们将目标检测分支和关系检测分支并行地输入到 Transformer 中，融合物体和关系的上下文，在 Transformer 的自注意力机制中，物体和物体、物体和关系、关系和关系之间的信息可以同时传递，形成一个更加灵活的推理过程。对于最后一个问题，我们使用交叉注意力机制从特征图中自适应地选择特征。我们的贡献可以总结如下：

1. 我们提出了一种基于 Transformer 的真正端到端的场景图生成网络。

2. 为了进一步减少成对组合的关系数目，我们提出使用交叉注意力机制将关系

的复杂度从$O(n^2)$降到$O(n)$。

3. 我们的方法目前已经可以接近主流网络模型的性能。

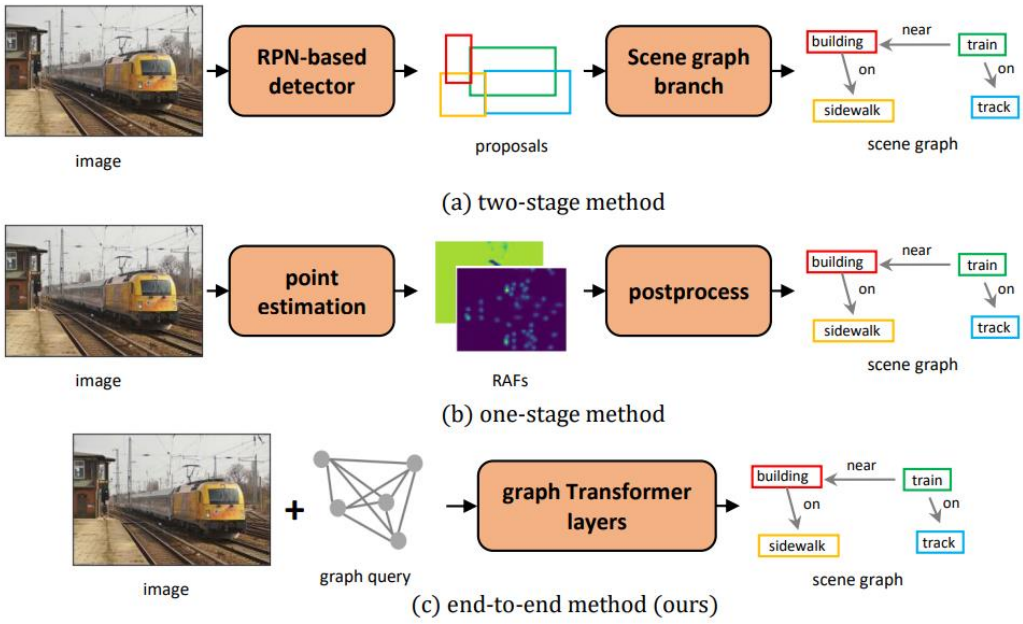


图 1：在两阶段方法中（图 1(a)），模型被设计成一种多阶段的模型，该类模型使用基于 RPN 的物体检测器去找到所有可能存在物体的候选框，并使用场景图子分支来去预测场景图的各个组件部分。在单阶段模型中（图 1(b)），该类模型将场景图生成看成关键点估计问题，他们的模型简洁且高效，但是这些模型需要诸如 RAF 人为构造的结构，并且在预测阶段需要复杂的后处理步骤。我们的模型（图 1(c)）是一个真正端到端的模型，它将图像和查询图作为输入，输出即得场景图，没有人为构造的组件或复杂的后处理步骤。

2 相关工作

场景图生成 现存的场景图生成方法[9,10,11,12,13,14]大多搭建在基于 RPN 的检测器上，它们首先利用 RPN 产生可能存在物体的候选框，从每个候选框中提取视觉特征，然后利用这些视觉特征构建成图结构，并应用几轮图节点间消息传递迭代，最后在预测阶段通过从视觉特征、空间关系特征和语义关系特征中提取线索对关系进行分类。 与这些两阶段方法，即先生成候选框再对关系进行分类的方法不同，[15]受人体姿态估计方法的启发，将场景图生成任务视作关键点估计，并行地对物体和关系进行定位和分类，与两阶段方法相比，他们的模型结构简单且显著缩短了推理时间，但是他们的方法仍然存在人为干预。我们的方法和这些方法不同，我们构建了一个真正端到端的网络，消除了诸如 RAFs[16]人工设计

的组件以及复杂的后处理操作，我们的网络结构非常简单，且网络输出即所得。

人物关系检测 人物关系检测是和场景图生成十分相关的任务，与场景图生成任务不同的是，人物关系检测是生成以人为中心的场景图，而且图像中的关系并不像场景图生成任务那样密集。这两项任务上有很多相似之处，许多观点相通。人物关系检测的方法也可以分成三个分支：两阶段方法[21,22,23,24]、单阶段方法[25,26,27]和端到端的[28,29]。实际上，针对人物关系检测已经存在端到端的模型[28,29,30,31]，但是他们的方法是针对人物关系三元组的，我们的方法可以覆盖更一般的情形，输入和输出是一个图结构。

卷积操作之外的图推理 为了捕获远程依赖关系和上下文环境，一些工作[32,33,34,35]尝试将图模型集成到他们的网络中，它们采用（1）从图像特征空间投影到语义符号空间，（2）在语义符号空间做符号推理，（3）从语义符号空间反投射到图像空间以细化原有特征三个步骤来做卷积操作之外的图推理。我们的方法也将这三个步骤结合到我们的模型中，但与使用 RPN 来限制特征区域的方法[36]不同，我们使用交叉注意力机制来自适应的选择性投影和反映射视觉特征。

结合外部知识的共识推理 有两种外部知识，一种是从大规模语料库中学习得到的词嵌入，他们隐式反映了词与词之间的关系。第二种是由人为构造的知识图谱，在这个图谱中，容纳了世界万物之间的关系，比如说 man 和 woman 共同输入 person 这个子集[19]。受这些工作[2,37]的启发，我们也尝试将外部知识容纳进我们的模型中，我们跟随[19]使用词嵌入和人为构造的知识图谱。

Transformer 在 CV 中的运用 Transformer[38]曾经在机器翻译中取得的很好的效果，近两年来，大量的工作尝试将 Transformer 应用到计算机视觉中的各个任务中，比如多模态学习[39,40,41]、语义分割[42]、图像分类[43]、目标检测[44]等。虽然已经存在将 Transformer 应用于场景图生成任务或视觉关系检测任务的方法[13,18,17]，然而，他们的方法使用 Transformer 来融合图节点之间的特征，并且他们的方法仍然是基于 RPN 的两阶段方法。我们的工作与他们不同，我们将跟随[43]去搭建一个真正端到端的模型，其中的检测分支是基于编码器-解码器的结构，而不是基于 RPN。

3 方法

如图 2 所示，我们的模型有三种类型的输入：（1）视觉特征序列，（2）场景图查询，（3）外部知识。将它们输入到由多层 Transformer 子层堆叠形成的子图融合模块，使得查询能够使用交叉注意力机制从视觉特征分支和外部知识分支提取场景图特征信息。我们的 Transformer 子层被设计成一种级联结构，一步步地产生场景图。在查询结果几轮特征聚集之后，会形成场景图特征序列，这些序列将会被去回归物体的位置和分类物体、关系的类别，从而可以构建出场景图。

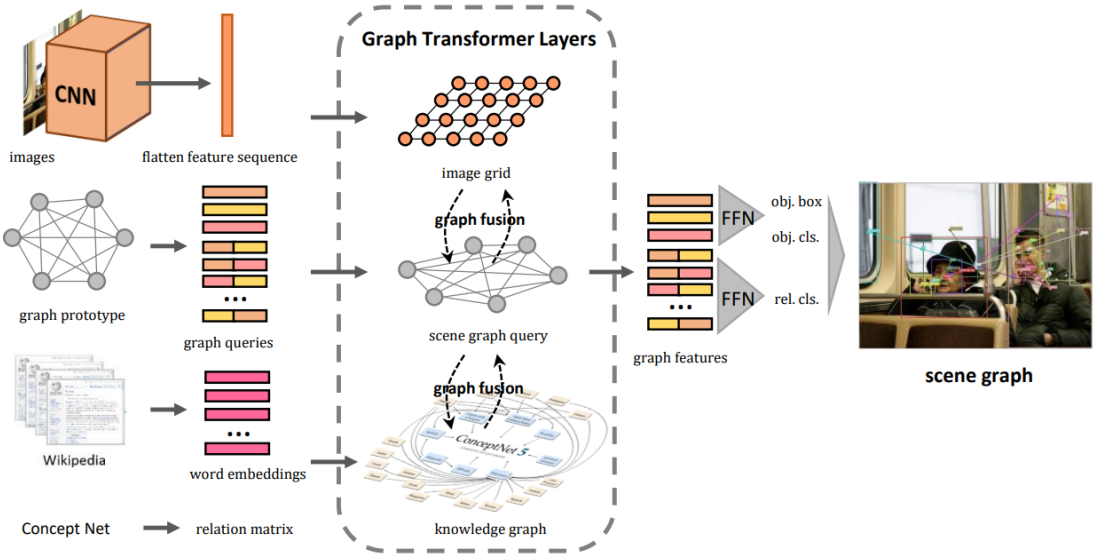


图 2：这张图展示了我们模型的主要结构。我们的模型有三种类型的输入：（1）视觉特征序列、（2）查询图序列和（3）外部知识。这些多元输入被编码成一个一维向量并被输入进 Transformer 层中以自适应地在图像特征网格图、查询图和有外部知识构建的知识图谱中进行特征交换和融合。在堆叠的 Transformer 子层中，查询图将会从另外两个分支中聚集信息。在数轮迭代之后，他们将被输入到 FFN 去回归物体的包围框和物体、关系的类别。

3.1 多源输入

视觉分支 给定形状为$\mathbb{R}^{w\times h\times 3}$的一张图片$I$，我们使用 CNN 去提取特征，得到 2D 特征图$F\in\mathbb{R}^{\frac{w}{s}\times\frac{h}{s}\times c}$，这里$s$是 CNN 的卷积总步长，$c$是输出特征图的维度。由于 Transformer 只能接受 1D 序列的输入，我们像大多数方法一样，将二维特征图展开成长度为n的一维特征序列$\{\mathbf{f}_i\}_{i=1}^{n}$。为了保留二维空间位置信息，我们仿照[43]使用由 sine 函数计算的位置嵌入[45]。

外部知识分支 我们跟随[19]使用词嵌入向量表示和先验关系来构建我们的知识图谱，词嵌入表示是从 GloVE 根据数据集中所有物体和关系的类别选择出来的，而先验关系是从 Concept Net 收集得到的。从形式上讲，我们将由外部知识所构建出的图记为$\mathcal{G}^S = \{\mathcal{V}^S, \mathcal{E}^S\}$。它有两个组成部分，即图节点集合$\mathcal{V}^S$和关系集合$\mathcal{E}^S$。图节点涵盖了数据集中的所有类别，拿 Visual Genome 数据集举例，该数据集中存在 150 个物体类别，那么该图中就存在 151 个物体图节点，新增的一个节点表示空物体，类似的还有 51 个关系图节点。这些节点使用词嵌入表示向量作为自己的语义特征。图中的边表示图节点以何种方式关联，这些关联方式在我们的实现中表示成不同层次的关系矩阵，存在联系的节点对在关系矩阵中对应位置置 1，否则置 0，我们推荐读者去[19]查看更多的细节。

图查询分支 图查询$\mathcal{G}^Q$是一个图，图中节点起初没有特定含义，这张图只表示初始场景图的拓扑结构，我们将这张图称之为场景图原型。这个场景图原型可以被视为是一种带有记忆功能的图，它保存了各种各样的场景图原型，就如同我们的大脑，即使我们闭上眼睛，我们的大脑仍然可以浮现出各种场景。我们固定了 100 个物体查询，并将所有对象成对连接起来，形成一个完全图。在这里，成对组合物体对于搭建一个真正端到端的模型是必要的，我们没有采用像[30]这样的更加复杂的关系查询，因为它会在推理中引入更复杂的后处理过程并且在训练阶段引入更复杂的损失计算。像[43]一样，我们将所有图查询初始化为一个零向量，并对每个查询附加一个在训练过程中学习得到的位置向量以区分图中的各个节点和保留拓扑结构。对于物体q_i^o，它学习得到的位置向量表示成p_i^o，对于物体查询节点q_i^o和物体查询节点q_j^o，位置向量通过下式计算

$$p_{ij}^r = \mathrm{FC}\left(\mathrm{Concat}\left(p_i^o, p_j^o\right)\right)$$

我们连接两个物体位置向量并使用全连接层是他们和物体位置向量的维度保持一致。

3.2 Transformer 模块

Transformer 图融合模块是由几个堆叠的 Transformer 子层构成，它的输入包括视

觉特征序列$\{\mathbf{f}_i\}_{i=1}^{n}$，场景图查询$\{\mathbf{q}_i\}_{i=1}^{m}$和带有语义信息的词嵌入表示向量序列$\{\mathbf{s}_i\}_{i=1}^{k}$，输出是场景图的特征。Transformer 模块是以一种级联的方式处理这些输入，如下式所示，

$$\{\mathbf{q}_i^{l+1}\}_{i=1}^{m} = \mathcal{G}\left(\{\mathbf{f}_i^{l}\}_{i=1}^{n}, \{\mathbf{q}_i^{l}\}_{i=1}^{m}, \{\mathbf{s}_i^{l}\}_{i=1}^{k}\right), l = 0,1,2,\dots,N \quad (1)$$

这里上标表示该向量序列所属的层级，N是 Transformer 子层的数目，$\mathcal{G}$表示 Transformer 模块中的一个子层，我们将在下面详细展开对它的介绍。

在 Transformer 子层中，我们使用交叉注意力机制在三个来源不同的输入做特征融合。如图#所示，信息流通包括四个方向：（1）从视觉分支流入图查询分支，（2）从图查询分支流入知识图谱分支，（3）从知识图谱分支流回图查询分支，（4）从图查询分支流回视觉特征分支。首先，我们使用交叉注意力机制让图查询向量序列$\{\mathbf{q}_i\}_{i=1}^{m}$去从视觉特征分支中聚集视觉信息$\{\mathbf{f}_i\}_{i=1}^{n}$。

$$\{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m} = \mathrm{CrossAttention}(\{\mathbf{q}_i\}_{i=1}^{m}, \{\mathbf{f}_i\}_{i=1}^{n}, \{\mathbf{f}_i\}_{i=1}^{n}) \quad (2)$$

这里$\{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m}$是一个展开来的场景图视觉特征，$\mathrm{CrossAttention}(Q,K,V)$是一个以$Q$为查询、以$K$为键、以$V$为值的多头注意力层。在特征信息从视觉分支流入场景图分支后，场景图查询中的每个向量有了它相对应的视觉含义。然后，我们把这些场景图视觉特征投影到使用先验知识构建的语义图中，

$$\{\mathbf{f}_i^{Q\to S}\}_{i=1}^{k} = \mathrm{CrossAttention}\left(\{\mathbf{s}_i\}_{i=1}^{k}, \{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m}, \{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m}\right) \quad (3)$$

这里$\{\mathbf{s}_i\}_{i=1}^{k}$是语义知识图谱中的节点的词嵌入表示向量，$\mathbf{f}_i^{Q\to S}$是语义图谱中融合视觉特征后的结点$i$的特征。正如第 3.1 节所示，这张语义图谱中的图节点由词嵌入表示向量组成，图中的边由两个关系的先验知识构成。在我们将带有视觉信息的场景图向量投影到该语义图谱中之后，我们希望在图片中出现的那些物体和关系在这张语义知识图谱中能够被激活，就如同我们人类看到一张图片后，图片中的内容将会在我们大脑中形成各种印象和概念。在这种感知过程之后，我们还会

结合我们头脑中的先验知识来分析图片中的物体是如何组织起来的。为了模拟这个过程，我们使用自注意力机制来在这张语义图谱中做图推导。

$$\{\mathbf{f}_i^{S\to S}\}_{i=1}^{k} = \text{SelfAttention}\left(\{\mathbf{f}_i^{Q\to S}\}_{i=1}^{k}, \text{weight} = M\right) \tag{4}$$

这里$\mathbf{f}_i^{S\to S}$是知识图谱中的节点i在自注意力机制构成从其他图节点收集特征之后的语义特征。我们使用注意力权重矩阵M来限制这张语义图上的信息传递过程，下面我们将展开介绍权重矩阵M的细节。注意到在这张语义图上，图节点表示概念，即数据集中表示物体的单词和表示关系的单词，这些图节点可以被分成m_o个物体图节点和m_r个关系图节点。拿 Visual Genome 数据集来说，我们将会有 202 个图节点，其中 151 个是物体图节点，51 个是关系图节点。正如前面所述的那样，语义图谱中的每个节点将词嵌入表示向量作为自己的语义特征，而注意力权重矩阵$M = (M_1, M_2, \ldots, M_k)$是一一系列关系矩阵，每个关系矩阵是一个二值矩阵，它表明这两个概念即单词之间是否相关而且以那种方式相关。这里有k个关系矩阵，每个矩阵都代表不同的含义，诸如 ISA 关系、USEFOR 关系、SUBSETOF 关系等等，这个矩阵是由[19]构建的，我们建议读者去[19]查看更多的细节。如方程#表示的那样，我们将这些表示两个单词之间是否存在关系的注意力权重矩阵应用到自注意力权重上面。在这背后的动机很简单，我们希望网络可以关联那些意义相关的事物，就如同我们在看到“racket”之后，脑海里会浮现出“tennis”、“player”、“hold”、“short”等概念。在这个过程之后，语义特征将被融入视觉特征和和它相关的概念特征。这些语义特征将会被反投射到查询图中。

$$\{\mathbf{f}_i^{S\to Q}\}_{i=1}^{m} = \text{CrossAttention}\left(\{\mathbf{f}_i^{V\to Q}\}_{i=1}^{m}, \{\mathbf{f}_i^{S\to S}\}_{i=1}^{k}, \{\mathbf{f}_i^{S\to S}\}_{i=1}^{k}\right) \tag{5}$$

至今，已经从另外两个分支收集到足够信息，初始为空的场景图查询向量将被赋予有意义的特征，这些特征取之于视觉图像和外部知识。为了去把握图片中物体与物体之间、物体和关系之间、关系和关系之间的依赖性，我们将自注意力机制应用到查询图中进行图推导

$$\{\mathbf{f}_i^{Q\to Q}\}_{i=1}^{m} = \text{SelfAttention}\left(\{\mathbf{f}_i^{S\to Q}\}_{i=1}^{m}\right) \tag{6}$$

这里$\{\mathbf{f}_i^{Q\to Q}\}_{i=1}^m$对应于方程 1 的$\{q_i^{l+1}\}_{i=1}^m$，在这个过程中，同一张图片中的物体和关系可以相互传递特征，并且上下文信息可以被散布到整张图的各个角落。查询图特征将会被用来去进行分类和回归，这将在章节 3.4 介绍。

最后我们将图查询特征反投影到视觉特征中去修正原先的视觉特征，

$$\{\mathbf{f}_i^{Q\to V}\}_{i=1}^n = \text{CrossAttention}\left(\{\mathbf{f}_i\}_{i=1}^n, \{\mathbf{f}_i^{Q\to Q}\}_{i=1}^m, \{\mathbf{f}_i^{Q\to Q}\}_{i=1}^m\right) \quad (7)$$

这里$\{\mathbf{f}_i^{Q\to V}\}_{i=1}^n$对应于方程 1 的$\{\mathbf{f}_i^{l}\}_{i=1}^n$。

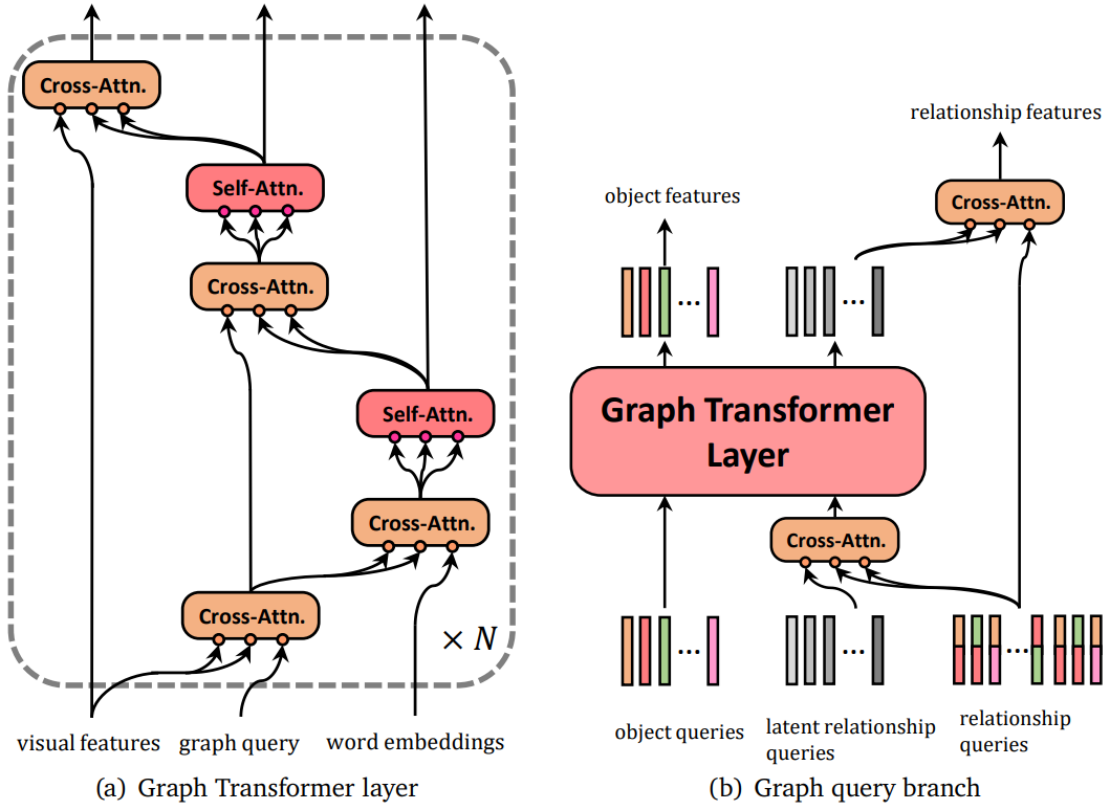


图 3：图 3(a)展示了数据量在 Transformer 子层的流动过程。图 3(b)展示了我们如何减少关系的规模。

3.3 减少完全查询图中关系的规模

我们在查询图中，列举了场景图中所可能出现的所有关系，当物体查询的数量增加时，关系查询的规模将会大到不可接受。为了解决这个问题，我们使用交

叉注意力机制来减少关系的规模。如图所示，我们定义了固定大小的隐含关系查询，这些隐含关系查询被初始化成零向量并且他们位置向量表示在训练中通过学习得到。隐含关系查询向量被输入进交叉注意力层中去从完全图中选择可能有具体含义的关系查询，之后的任何复杂操作，均附加在这些隐含关系查询向量中。最后这些隐含关系查询将会被反投影到原来的关系查询向量中。以这样的方式，我们把关系查询的复杂度从$O(n^2)$降到$O(n)$，与此同时没有破坏完全图结构。

3.4 推理和损失

在场景图查询从视觉分支和语义分支收集到足够多的特征之后，这些特征将会被用于分类和回归。对于物体查询来说，使用单 MLP 层分类，使用三隐层 MLP 回归，即

$$\mathbf{b}_i = \mathrm{MLP}\left(\mathbf{f}_i^{Q\to Q}\right) \text{ and } c_i^o = \mathrm{MLP}\left(\mathbf{f}_i^{Q\to Q}\right) \tag{8}$$

这里$b_i \in \mathbb{R}^4, c_i^o \in \mathbb{R}^{C_o+1}$，$C_o$是数据中所有物体类别的数量。对于关系查询来说，他们仅仅被用于分类，由于和关系相关联的物体已经隐含在关系的位置向量上，这里不必再去对关系的位置做回归，因此我们有

$$c_{ij}^r = \mathrm{MLP}\left(\mathbf{f}_{ij}^{Q\to Q}\right) \tag{9}$$

这里$c_{ij}^r \in \mathbb{R}^{C_r+1}$是物体$i$和物体$j$之间的关系，$C_r$是数据集中所有关系类别的数量。在推理阶段中，首先把空物体查询排除，并且只保留与关系关联的两个物体均不为空的关系，利用这些物体的位置、类别和关系的类别构建出场景图，注意到这里没有引入任何复杂的后处理步骤。

在训练阶段我们使用 Hungarian 二分匹配[43]来将标记赋给每一个预测，损失通过下式计算

$$\mathcal{L}_{\text{total}} = \lambda_{\text{L1}}\mathcal{L}_{\text{L1}} + \lambda_{\text{IOU}}\mathcal{L}_{\text{IOU}} + \lambda_{\text{CLS}}\mathcal{L}_{\text{CLS}} + \lambda_{\text{knowledge}}\mathcal{L}_{\text{knowledge}} \tag{3-11}$$

这里$\mathcal{L}_{L1} = \sum_i \left\|\mathbf{b}_i - \hat{\mathbf{b}}_i\right\|_1$是预测候选框$\mathbf{b}_i$和标记候选框$\hat{\mathbf{b}}_i$之间的$l_1$损失，$\mathcal{L}_{\text{IOU}}$是广式 IoU 损失。我们使用交叉熵损失来计算$\mathcal{L}_{\text{CLS}}$。为了避免模型过于倾向于把查询特征分类成空物体和空关系，我们对空关系和空物体设置了一个权重。与[43]相比，我们新增了两个损失，一个用来计算关系分类的损失，另外一个损失用来对知识投影进行惩罚。

我们把预测关系分支的输出视作一个大小为(n_{obj}, n_{obj})的二维矩阵，这里n_{obj}表示物体查询的数目。在这个矩阵中，位置(i,j)是一个逻辑概率向量$c_{ij}^{r} \in \mathbb{R}^{C_r+1}$，它代表这物体$i$和物体$j$之间的关系被分成各类的概率。我们根据标记和预测结果二分匹配的结果来构建真值矩阵，这个真值矩阵中的每一个位置(i,j)被赋值成图查询节点i和图查询节点j所对应的物体之间的关系标签。构建完成这两个矩阵之后使用带有对空关系的惩罚权重为 no_rel_weight 的交叉熵损失来计算关系的损失。为了进一步区分已标记标签和未标记标签，我们对关系设置了两个损失，第一个损失就如同上述的那样使用权重为 no_rel_weight 的交叉熵损失来计算预测关系矩阵和真值关系矩阵之间的差距。我们还添加了对已标记标签的辅助损失，对于那些数据集已经标注的标签，使用权重 aux_no_rel_weight 来计算交叉熵损失。

至于知识投影损失，首先把物体特征和关系特征使用单 MLP 层投影到一系列具有不同含义的隐语义空间中。在每个语义空间中，每一个物体和关系都被赋予一个语义标签，这个语义标签有外部知识所定义。例如，在表示“SIMILAR WITH”含义的隐语义中，由于在外部知识中“wearing”和“has”、“wears”表达同一个含义，那么标签为“wearing”相对应的查询被分类成“has”和“wears”多标记。在这背后的动机是，我们希望带有相同含义的单词应该被投影在相同的语义空间中。

4 实验

4.1 实现细节

跟随[43]使用 ResNet-50 来提取特征，特征图由 ResNet-50 输出且输出特征的通道数为 2048，我们将它下采样到 256。我们固定了维度为 256 的 100 个物体查询和 100 个隐含关系查询。在 Transformer 层中，使用 5 层 Transformer 子层，每一层使用 8 个并行的多头注意力层，前馈特征的维度为 2048。在损失函数中，设置λ_{L1}为 5，λ_{IOU}为 2 并且λ_{CLS}为 1，空物体权重设置为 0.1，空关系权重设置为 0.01。使用 AdamW 优化器来训练模型，初始学习率设置为5×10^{-5}，并且在 60

轮训练中衰减为原来的 0.1。训练之初加载 ResNet-50 并冻结批归一化层，我们使用 4 卡进行训练，批大小设置成 8，整个训练过程大概消耗 4 天。

4.2 数据集

我们使用 Visual Genome 数据集[46]来测试模型场景图生成的性能并使用 HICO-DET 数据集来测试模型视觉关系检测的性能。原 Visual Genome 数据集包含 108, 077 张图片，我们使用[48]的划分来训练和验证我们的模型，在这个划分中，去除了出现频率不高的物体和关系，只留下 150 种物体类别和 50 种关系类别。在这个数据集中，我们测试带有图限制或不带图限制的 recall@20/50/100 指标和平均 recall@20/50/100 指标。HICO-DET 数据集中包含了 47, 776 张图片，其中 38,118 张用于训练模型，9,658 张用于验证模型。我们使用 mAp 作为自己的验证指标。

Recall @ K / No-graph Constraint Recall@K		Scene Graph Detection											
		R@20/50/100			ng-R@20/50/100			mR@20/50/100			ng-mR@20/50/100		
External Knowledge	VCTree[51]	22.0	27.9	31.3				5.2	6.9	8.0			
	KERN[52]		27.1	29.8		30.9	35.8		6.4	7.3			
	GPS-NET[53]	22.3	28.9	33.2						9.8			
	MOTIFS-TDE[50][54]	12.4	16.9	20.3				5.8	8.2	9.8			
	GB-NET[19]		26.3	29.9		29.3	35.0		7.1	8.5		11.7	16.6
	RelDN[56]	21.1	28.3	32.7		30.4	36.7						
Visual Only	VTransE[57]		5.5	6.0									
	FactorizableNet[58]		13.1	16.5									
	IMP[48][50]	14.6	20.7	24.5									
	Pixels2Graphs[59]					9.7	11.3						
	Graph R-CNN[49]		11.4	13.7									
	VRF[60]		13.2	13.5									
	CISC[36]	7.7	11.4	13.9									
	HRNet[15]	16.1	21.3	25.1	16.7	23.5	29.2	2.7	3.6	4.2	3.8	5.7	7.5
	CMAT[15]	22.1	27.9	31.2	23.7	31.6	36.8						
	KERN[14]		27.1	29.8		30.9	35.8		6.4	7.3			
	LSBR[61]	23.6	28.2	31.4	26.9	31.4	36.5						
	SGGTR (Ours)	18.7	24.7	28.9	20.5	28.1	33.5	4.0	5.9	7.4	5.8	9.7	13.7

表 1：在数据集 VG-150 的召回率验证结果

4.3 和其他模型性能比较

我们收集了近四年在 VG 数据集和 HICO-DET 数据集上的结果。正如表 1 和表 2 所示，我们的模型正在接近主流模型的性能，但是仍然存在不可忽视的差距。在 Visual Genome 数据集中，我们的模型达到了 18.7 的 recall@20 并且可以打败诸如 Graph RCNN[49]、CISC[36]、IMP[48,50]等模型。但是和最近刚刚发表的模型仍然存在差距。在 HICO-DET 数据集上，和主流模型也存在几个 mAp 值的下降。

Method		mAp		
		full	rare	non-rare
Two-stage	CHGN[63]	17.57	16.85	17.78
	DRG[64]	24.53	19.47	26.04
	VCL[23]	23.63	17.21	25.55
	ATL[64]	23.81	17.43	24.32
One-stage	PPDM[25]	21.94	13.97	24.32
	IP-Net[26]	19.56	12.79	21.58
	GGNet[27]	23.47	16.48	25.60
	AS-Net[31]	28.87	24.25	30.25
	HOTR[30]	25.10	17.34	27.42
End-to-end	HoiTransformer[38]	26.61	19.15	28.84
	QPIC[29]	29.90	23.92	31.69
	SGGTR(ours)	21.94	17.14	23.37

表 2：在数据集 HICO-DET 的结果

4.4 消融实验

为了进一步验证模型的性能在添加额外的损失之后能否进一步被提高。我们使用带有在 3.4 节描述的知识损失和辅助关系损失来训练 SGGTR。结果如表 3 所示，在添加了这两个损失之后，召回率在验证集下降地很厉害。但是，模型在训练集上提高了不少。注意到我们在 VG 数据集上训练我们的模型的时候，没有使用任何的数据增强技巧。我们把这样的结果归咎于过拟合。我们将会再次使用数据增强重新训练我们的模型，结果将会在后续的工作展出。我们也发现，基于 DETR 的 SGGTR 模型在训练集上的物体检测指标非常高，这说明基于编码解码

器的结构对于物体检测是关键的。对比基于 DETR 的模型和 SGGTR 模型，SGGTR 模型要略微比 DETR 模型稍好一些，这说明投影-推理-反投影适合于处理图数据。

Model	w/o knowledge loss	w/o auxiliary relationship loss	on eval set				on training set			
				recall@				recall@		
			mAp	20	50	100	mAp	20	50	100
DETR-Based	✗	✗	23.61	13.47	20.93	25.90	53.71	31.99	43.69	51.87
SGGTR	✗	✗	24.50	18.45	24.30	28.56	34.46	24.56	29.96	31.24
	✗	✓	24.11	11.05	16.94	22.01	36.39	39.07	48.63	54.98
	✓	✓	22.32	10.01	15.20	20.23	32.78	35.88	44.20	49.64
SGGTR with exknowledge	✗	✗	22.36	14.66	21.82	26.28				
	✓	✓	21.99	9.96	15.35	20.19	43.53	41.75	52.06	58.60

表 3：在数据集 Visual Genome 的消融实验

我们还在 HICO-DET 数据集上对比了 SGGTR 和 DETR，正如表 4 所示，SGGTR 的结果已经和 DETR 的结果非常接近。并且通过对比带有 20 个物体查询和带有 50 个物体查询的 SGGTR，性能几乎没有太大的差别，这说明模型结构和物体查询数量并不是影响我们模型性能的决定性因素，使我们的方法比主流方法差的原因可能是由于损失的设置和查询图的设计。

Model	Object query number	Relationship query number	Graph Transformer layer number	mAp		
				full	rare	non-rare
DETR	20	20	5	22.31	18.36	23.49
SGGTR	20	20	5	21.93	18.45	22.97
	20	20	6	20.51	15.68	21.96
	50	50	5	21.94	17.14	23.37

表 4：在数据集 HICO-DET 的消融实验

4.5 分析

为了分析模型是否像我们所期待的那样工作，我们可视化了多头注意力层中的注意力权重。如图 4 所示，我们发现查询图在查询到视觉特征的注意力会聚焦在物体的边缘，这就和[43]中的 decoder 类似。但是，在视觉到查询图的注意力中，场景图的查询节点向量被散布在整张图中，并没有聚焦在有意义的目标区域。

这说明如果查询图特征被反投影到如图 4(b)所示的关键区域之后，查询图将会影响到视觉特征。在查询图到查询图的自注意力权重矩阵中，我们观察到消息在相关的物体中传递，例如图 4(c)中的 bus 和 wheel。我们将会把更多可视化的例子放在附录。

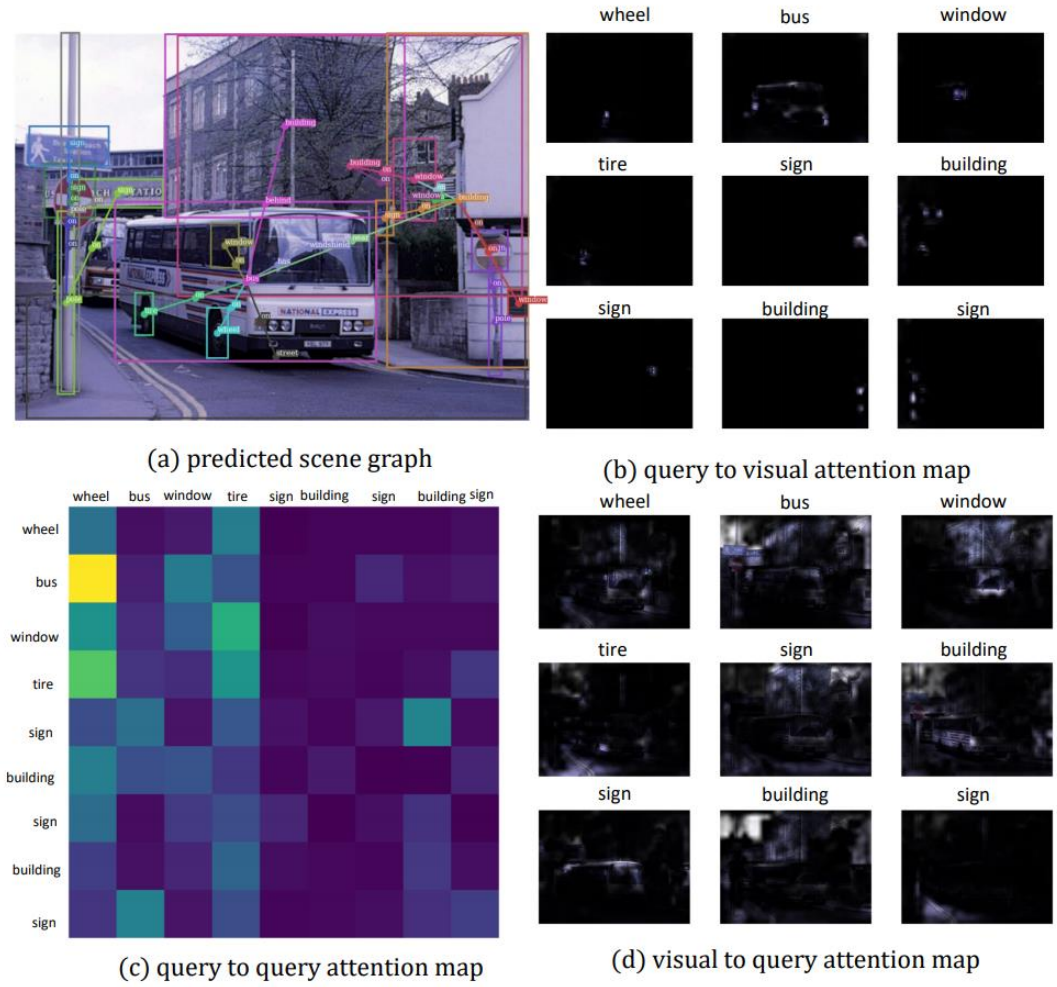


图 4：我们在 Visual Genome 数据集上可视化了注意力权重矩阵。图 4(a)使有我们模型所产生的场景图。图 4(b)使查询图和视觉特征的交叉注意力权重图。图 4(c)是查询图的自注意力权重矩阵。图 4(d)是视觉特征和查询图特征之间的注意力热图。

我们也在 HICO-DET 数据集上可视化了结果，正如图 5 所示，由于我们使用了潜在关系查询向量来搜索关系的视觉依据，所以很难去追溯关系注意到哪些区域，这里我们仅仅可视化了一些看起来比较有道理的热力图。

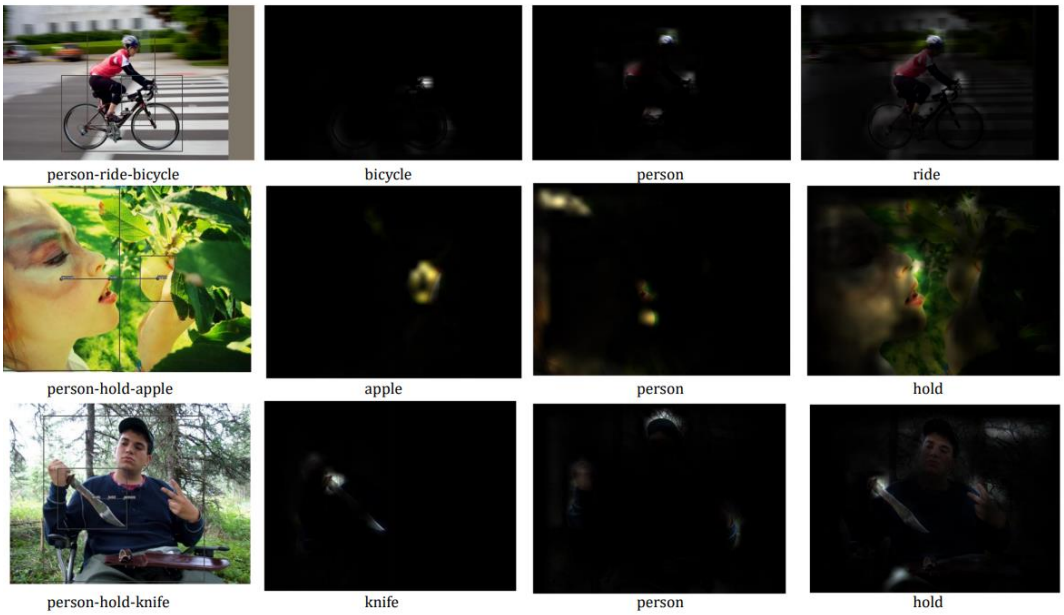


图 5：在 HICO-DET 数据集上的可视化的注意力图

图 6 展示了一些有我们模型生成的场景图，生成的场景图是有道理的。然而，在我们检查由我们模型输出的场景图时，也发现了一些失败的例子，这些将会被展示在附录中。正如[54]所说的那样，我们的模型存在和大多数模型相同的问题即偏倚问题。我们模型另外一个问题是我们的模型很难处理一些带有语义含义的关系，例如 HICO-DET 数据集中的 inspect、direct、load、load 等。

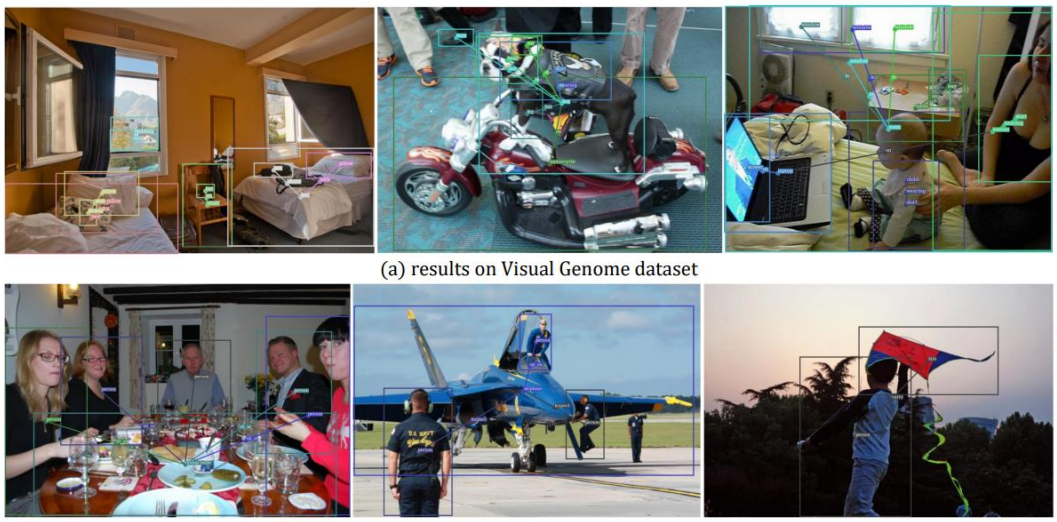
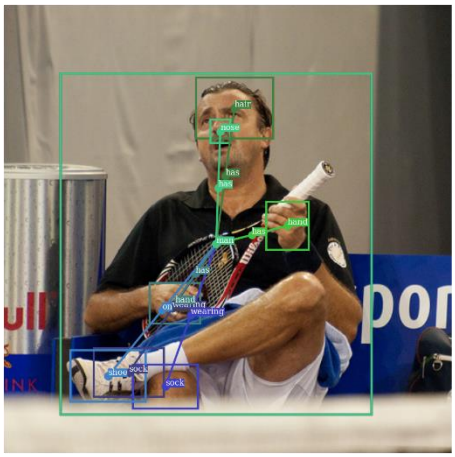
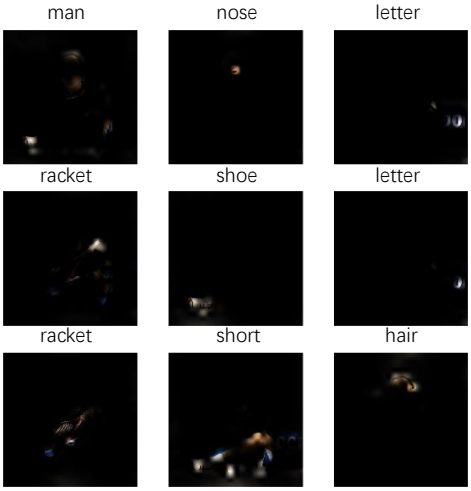


图 6：在 Visual Genome 数据集和 HICO-DET 数据集上的生成的场景图

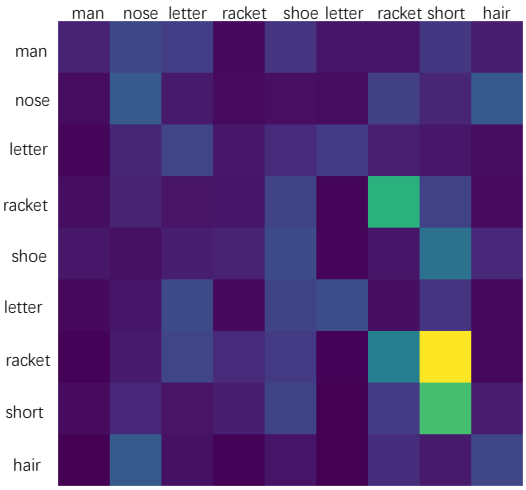
附录 3 更多可视化的结果



(a) predicted scene graph

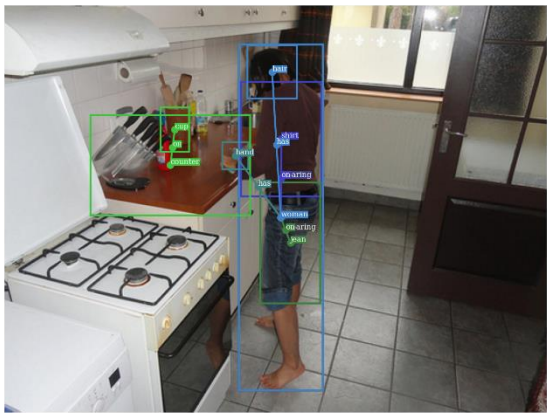


(b) query to visual attention map



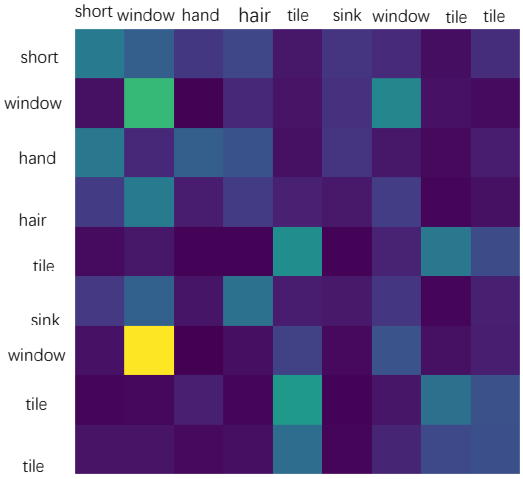
(c) query to query attention map

(d) visual to query attention map

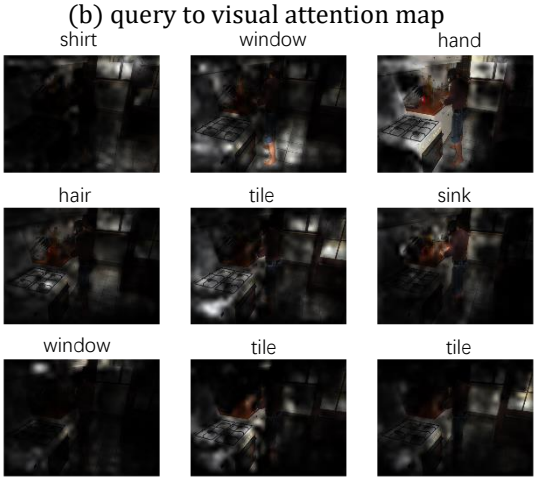


(a) predicted scene graph

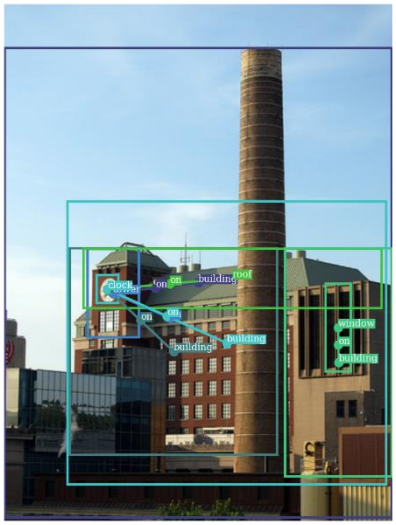
(b) query to visual attention map



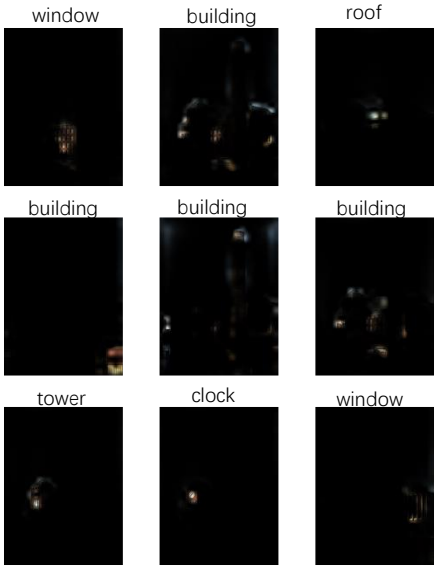
(c) query to query attention map



(d) visual to query attention map



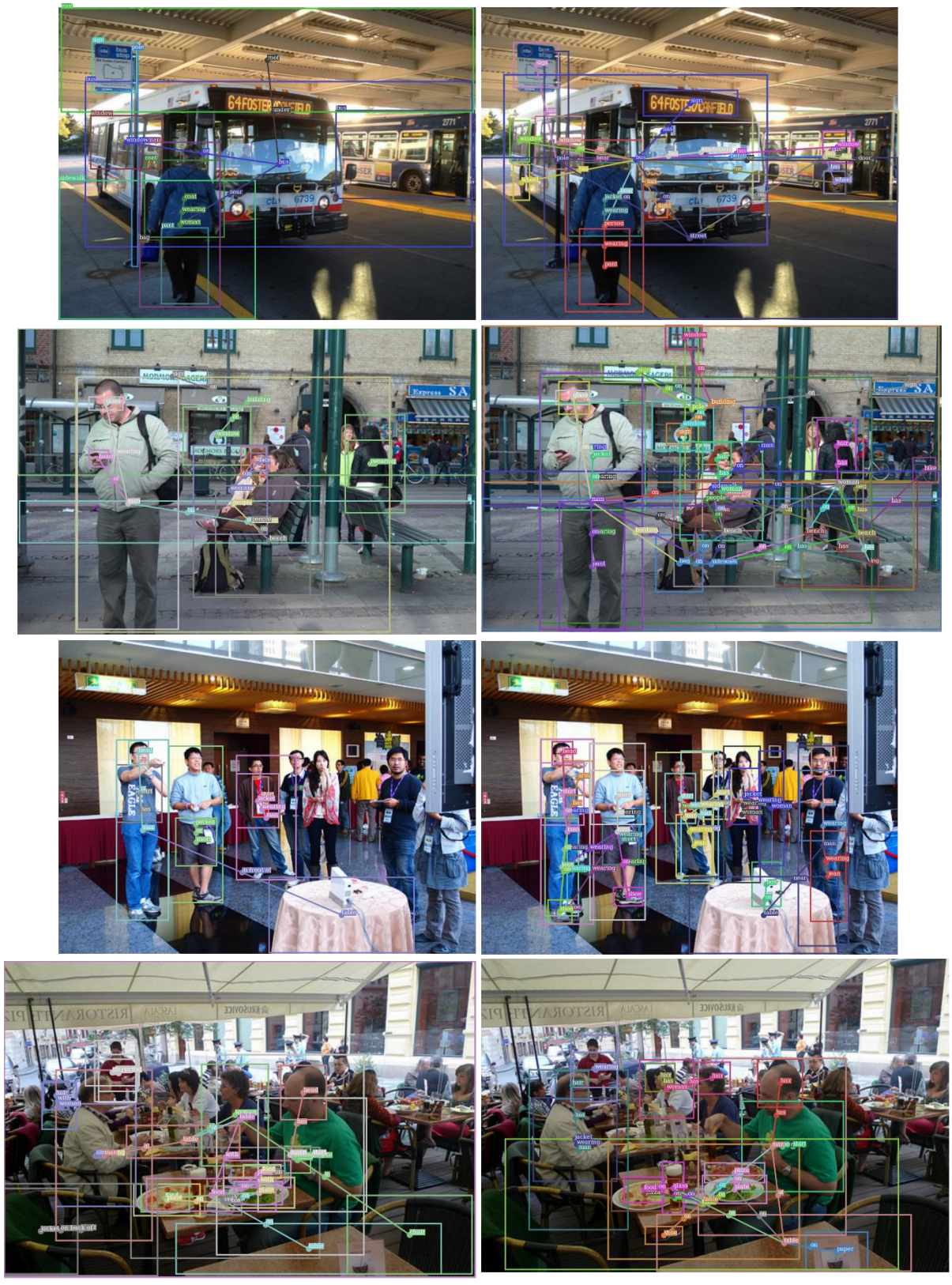
(a) predicted scene graph



(b) query to visual attention map

(c) query to query attention map

(d) visual to query attention map



(a) ground-truth (b) predicted results