基于外部先验和自先验注意力的图像描述生成方法

摘要

图像描述是一种结合计算机视觉和自然语言处理的跨模态任务，旨在理解图像内容并生成恰当的句子。现有的图像描述方法通常使用自注意力机制来捕获样本内的长距离依赖关系，但这种方式不仅忽略了样本间的潜在相关性，而且缺乏对先验知识的利用，这导致生成内容与参考描述存在一定差异。针对上述问题，文中提出了一种基于外部先验和自先验注意力 (External Prior and Self-prior Attention，EPSPA) 的图像描述方法。其中，外部先验模块能够隐式地考虑到样本间的潜在相关性进而减少来自其他样本的干扰信息。同时，自先验注意力能够充分利用前层的注意力权重来模拟先验知识，使其指导模型进行特征提取。在公开数据集上使用多种指标对 EPSPA 进行评估，实验结果表明该方法在保持低参数量的前提下能够表现出优于现有方法的性能。

关键词：图像描述；自注意力机制

1 引言

图像描述是同时涉及计算机视觉 (Computer Vision,CV) 与自然语言处理 (Natural Language Processing,NLP) 两个不同领域的任务，它是多模态的研究热点之一。该任务需要运用适当的词汇和语法来对图像进行描述，在帮助人们更好地理解图像🄴为视觉障碍者提供辅助服务和提高搜索引擎效率及环境交互等方面有着广泛应用。

尽管图像描述领域发展迅速，但仍面临着以下挑战：1) 需要克服图像与文本的异构性，实现跨模态信息的提取和对齐；2) 图像中存在大量显式和隐式的视觉语义信息，需要合理使用这些信息，并有针对性地生成描述；3) 需要根据上下文或先验知识进行推理，进而生成流畅的描述。

近年来，受机器翻译模型 [1] 的启发，大多数图像描述生成方法普遍采用编码-解码架构。文献 [2] [3] [4] 先利用卷积神经网络 (Convolutional Neural Network，CNN) 对输入图像进行编码，再利用循环神经网络 (Recurrent Neural Network，RNN) 来建模并生成一系列单词输出。然而，这种方法使得模型难以捕捉到长距离的上下文信息，并且在生成长文本描述时较为困难。随后，文献 [5] [6] 证明了基于自注意力机制的 Transformer 在 CV 和 NLP 领域拥有巨大潜力。得益于自注意力机制强大的长距离关系建模能力，基于 Transformer 框架的方法逐渐成为图像描述领域的主流方法。目前，基于自注意力机制的图像描述生成方法已经取得了很多研究成果。例如，Huang 等 [7] 提出 AoA(Attention on Attention) 模块，该模块通过扩展传统的注意力机制来确定注意力结果和查询结果之间的相关性。此外，Cornia 等 [8] 则利用额外的记忆向量来学习和编码先验知识，并在解码阶段使用多层连接来融合低级和高级

特征。上述方法虽然取得了不错的效果，但仍存在一些问题尚未解决。一方面，自注意力机制在计算时只考虑到样本内部的特征信息，却忽略输入样本与其他样本之间的潜在相关信息，而这类不同样本之间的联系对特征提取有着至关重要的影响。例如，当前样本存在一些噪声或冗余信息时，模型可以利用其他样本中包含的相关信息，来减少当前样本中的干扰，达到增强模型泛化能力的效果。另一方面，在缺少先验知识引导时，自注意力机制可能会导致模型难以有效地聚焦于重要的特征。但现有方法在利用先验知识时往往需要大量的数据集和额外的内存开销。

针对上述问题，本文在 Cornia 等 [8] 在 2020 年 CVPR 上发表的论文的基础上进行改进，提出了一种基于外部先验和自先验注意力的图像描述生成方法 (Extern-al Prior and Self-prior Attention，EPSPA)。该方法在编码端提出了外部先验模块 (External Prior Module，EPM) 和自先验注意力模块 (Self-prior Attention，SPA)。在解码端使用经典的 Transformer 解码器。具体来说，为捕捉到不同样本之间的潜在相关性，本文借鉴简洁高效的多层感知机 (Multilayer Perceptron，MLP) 设计了 EPM。与自注意力机制需要计算多个权重矩阵相比，EPM 由投影和静态参数化的循环矩阵组成，具有较低的参数量和计算复杂度。此外，本文在原有自注意力机制的基础上，在计算时将前层的注意力权重向后传递，进而提出了一种新的自注意力机制，本文称之为自先验注意力模块。

2 相关工作

很多研究者针对图像描述领域进行了诸多探索，取得了显著进展。目前，主流的图像描述方法都是基于编码器-解码器架构。本文将从基于 CNN-RNN 的图像描述，基于 Transformer 的图像描述，以及给本文提供灵感的 MLP 在视觉特征提取中的应用 3 方面来进行介绍。

2.1 基于 CNN-RNN 的图像描述

近年来，CNN 和 RNN 的方法在图像描述领域取得了显著进展。这类方法通常采用 CNN 模型来提取图像特征，然后将其输入到 RNN 模型生成描述语句。例如，Vinyals 等 [2] 提出了一个端到端的神经网络模型，使用 Inception-V3 [9] 作为 CNN 模块，使用长短期记忆网络 (Long Short-term Memory Network，LSTM) 作为 RNN 模块，并引入了注意力机制来动态选择图像区域进行描述。Xu 等 [10] 则使用 ResNet-101 [11] 作为 CNN 模块，并在 RNN 模块使用自适应注意力机制 [12]，以便在生成每个单词时同时考虑图像和文本信息。此外，还有一些工作尝试使用更复杂的 RNN 结构或引入其他信息来提高图像描述的质量和多样性。例如，Anderson 等 [13] 使用自下而上的注意力机制来预先检测出图像中的物体，并将其编码为特征向量，然后选择相关的物体生成描述。

2.2 基于 Transformer 的图像描述

Transformer [5] 是一种基于自注意力机制的神经网络架构，最初用于自然语言处理任务，如机器翻译和文本摘要。近年来，Transformer 也被广泛应用于图像描述任务，即根据给定的图像生成一段描述性文本。它由编码器和解码器组成，编码器将图像分割成若干区域，并提取每个区域的特征向量，解码器则根据编码器的输出和已生成的单词来预测下一个单词。Transformer 可以利用多头自注意力机制捕捉图像和文本之间的关系，从而生成更准确和流畅

的描述。目前，基于 Transformer 框架的图像描述已经有很多研究。例如，Herdade 等 [14] 在编码器中加入对象之间的空间关系进一步提取图像特征；Cornia 等 [8] 提出多层的网格状连接来融合低级和高级的图像特征，尽管基于 Transformer 的图像描述方法取得了巨大成功，但也存在不足。具体表现在 Transformer 的训练需要大量计算资源和数据资源。例如，在处理长序列时，需要计算所有位置之间的注意力权重，这会使得计算复杂度呈二次增长，从而带来巨大的计算开销，可能导致模型过拟合，同时还存在模型在不同领域和场景下的适应性问题。

2.3 MLP 在视觉特征提取中的应用

近期，多层感知机 (Multilayer Perceptron，MLP) 在视觉任务上取得了令人瞩目的成果，引发了学术界的广泛关注。最近一些工作表明，只要合理地设计输入输出和层间交互方式，MLP 可以在图像分类任务上达到与 CNN 或 Transformer 相媲美甚至超越的性能。例如，Tolstikhin 等 [15] 提出了 MLP-Mixer，它将图像划分为多个块，并将每个块映射为一个向量。然后，它使用两种类型的 MLP 层来处理这些向量: 一种是沿着空间维度进行交互的 channel-mixing MLPs，另一种是沿着通道维度进行交互的 token-mixing MLPs。这样就可以实现跨空间位置和跨通道特征的信息融合。其他工作也探索了如何利用 MLP 来实现高效且强大的视觉分类模型。例如，Liu 等 [16] 提出了 gMLP，它使用门控线性单元 (gated linear unit) 来替换传统的激活函数，并引入了空间感知门控模块 (spatially aware gating module)，可以根据不同位置调节信息流动。

3 本文方法

3.1 本文方法概述

文中表示现有方法对图像描述上下文的利用仍未得到充分探索。该架构改进了图像编码和语言生成步骤：它学习了图像区域之间关系的多级表示，并集成了已学到的先验知识，并在解码阶段使用网状连接来利用低级和高级特征。如图 1所示，主要的创新点有两个，一是在编码端提出了一个记忆增强注意力。将图像区域和其关系编码为一个多层次结构，其中考虑了低层次和高层次关系。通过 memory vector 实现。这个 memory vector 就是图中白色的那部分，主要是用来编码先验知识。二是在解码端句子的生成采用多层次结构，利用低层次和高层次的视觉关系。这是通过一个门控机制来实现的，该机制在每个阶段加权多层次的贡献。图中也可以看出编码器和解码器层之间创建了一个网格连接

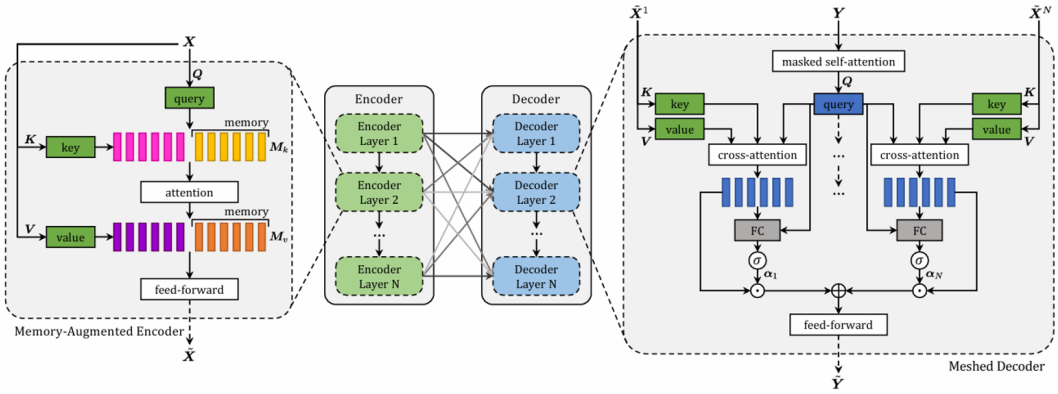

图 1. 方法示意图

3.2 记忆增强注意力模块

为了克服自我注意的限制，我们提出了一种记忆增强注意力。具体如下所示，在文中的方法中，用于注意力计算的 K 和 V 扩展了额外的“插槽”，这些“插槽”可以编码先验信息。

$$\begin{aligned} \mathcal{M}_{\text{mem}}(\boldsymbol{X}) &= \text{Attention}(W_q\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{V}) \\ K &= [W_k X, M_k] \\ V &= [W_v X, M_v], \end{aligned} \tag{1}$$

其中，M_k 和 M_v 是可以学习的参数矩阵，同时通过加入可学习的键和值，就有可能检索到尚未嵌入 X 中的学习知识。同时，上面的公式使查询集保持不变。

3.3 网格化解码器

文中的解码器以先前生成的单词和区域编码为条件，并负责生成输出标题的下一个标记。在这里，我们利用了上述输入图像的多级表示，同时仍然构建了一个多层结构。为此，我们设计了一个网格化注意力算子，它与原始 Transformer 中的交叉注意力运算类似，可以在句子生成过程中占据所有编码层的优势。

$$\mathcal{M}_{\text{mesh}}(\tilde{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \odot \text{Attention}\left(\boldsymbol{Y}, \tilde{\boldsymbol{X}}^i, \tilde{\boldsymbol{X}}^i\right) \tag{2}$$

其中 α_i 是与交叉注意力结果大小相同的权重矩阵。既调节每个编码层的单一贡献，也调节不同层之间的相对重要性。

4 复现细节

4.1 与已有方法理论对比

与原文中的方法相比，我做了一些轻量化改进。经典的自注意力机制仅能考虑到样本内部之间的联系，却忽略了样本间的潜在相关性。例如，两张图像可能具有相同的主题、颜色、纹理、形状等特征，这些特征可以构成图像之间的潜在相关性。自注意力机制可能会过度关

注某些特定样本中的噪声信息而缺乏对其他样本中全局特征的关注。并且在计算过程中，它需要存储所有位置的嵌入表示和对应的注意力权重，这会导致占用大量内存。

为此，本文提出具有线性复杂度的外部先验模块 EPM。它与经典自注意力机制的区别在于它不使用相似度作为权重矩阵，而是采用一个循环矩阵让模型从原始数据中学习权重。由于它只需要存储一个固定大小的矩阵和序列的嵌入表示，这能显著减少内存的占用。循环矩阵是由一组固定权重循环构成，矩阵中的每一行都是上一行循环右移得到的。相比于普通权重矩阵，循环权重矩阵的优势在于它可以捕捉样本间的依赖关系。这种循环性质使得模型能够在处理当前图像的特征时，同时参考前面图像的特征，减少当前干扰信息的影响，有助于模型理解不同图像之间的联系。

$$h_i = W_i \,\mathrm{Norm}\,(x_i) + b_i \tag{3}$$

$$\mathrm{EPM(X)} = Concat\,(h_1, h_2, \ldots h_H) \tag{4}$$

为了对模型进行引导，使注意力相关性更强，本文提出了自先验注意力模块 SPA。具体来说，当前编码层在进行计算时会用到前层的注意力权重作为参数补充，并传递到后一层。这使得注意力能关注到更相关的信息，同时对模型起到一定地指导作用。值得注意的是，本文为了减少冗余的计算，在自先验注意力模块中没有像传统方法一样使用多头，而是仅仅使用单头将特征映射到一个空间中，让模型更专注于当前特征。其操作表示为:

$$\begin{aligned} \mathrm{SPA}(Q, K, V,\ \mathrm{prior}\ ) &= \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d}} +\ \mathrm{prior}\ \right) V \\ \mathrm{prior}\ \ &= \mathrm{Softmax}\left(\frac{Q_{\mathrm{front}}\ {K_{\mathrm{front}}}^T}{\sqrt{d}}\right) \end{aligned} \tag{5}$$

其中 SPA 代表自先验注意力，prior 中的 Q_{front} 和 K_{front} 来自于前一层，第一层的 prior 初始化为空。自先验注意力在当前注意力权重的基础上加入之前的权重，然后像经典注意力一样进行加权求和。

4.2 实验环境搭建

实验环境搭建主要参考原文给出的 github 网址，按照其中的 README 文件一步一步搭建即可。github 网址如下：https://github.com/aimagelab/meshed-memory-transformer

4.3 与已有开源代码对比

在原始代码的基础上进行了改进，改进点已经在下图用红色箭头标出，首先是把原文的注意力机制更换为更轻量的自先验注意力，然后再加入了外部先验模块 EPM。

```
class EncoderLayer(nn.Module):
    def __init__(self, d_model=512, d_k=64, d_v=64, h=8, d_ff=2048, dropout=.1, identity_map_reordering=False,
                 attention_module=None, attention_module_kwargs=None):
        super(EncoderLayer, self).__init__()
        self.identity_map_reordering = identity_map_reordering
        self.mhatt = MultiHeadAttention(d_model, d_k, d_v, h, dropout, identity_map_reordering=identity_map_reordering,
                                        attention_module=attention_module,
                                        attention_module_kwargs=attention_module_kwargs)
        self.pwff = PositionWiseFeedForward(d_model, d_ff, dropout, identity_map_reordering=identity_map_reordering)

    def forward(self, queries, keys, values, attention_mask=None, attention_weights=None):
        att = self.mhatt(queries, keys, values, attention_mask, attention_weights)
        ff = self.pwff(att)
        return ff
```

```
class EncoderLayer(nn.Module):
    def __init__(self, d_model=512, d_k=64, d_v=64, h=8, d_ff=2048, dropout=.1, identity_map_reordering=False,
                 attention_module=None, attention_module_kwargs=None):
        super(EncoderLayer, self).__init__()  #super().__init__()的作用也就显而易见了，就是执行父类的构造函数，使得我们能够调用父类的属性
        self.identity_map_reordering = identity_map_reordering

        # self.mhatt = MultiHeadAttention(d_model, d_k, d_v, h, dropout, identity_map_reordering=identity_map_reordering,
        #                                 attention_module=attention_module,
        #                                 attention_module_kwargs=attention_module_kwargs)
        # self.mhatt =AFTLocal1(max_seqlen=50,dim=512)
        # self.attn = AFTLocal(max_seqlen=50,dim=512)
        self.attn = Attention(d_model, d_model, d_k, causal=True) if exists(d_k) else None
        self.mhatt  = EPM(d_model, 50, causal=True, act = nn.Identity(),heads=h, circulant_matrix = True)
        self.pwff = PositionWiseFeedForward(d_model, d_ff, dropout, identity_map_reordering=identity_map_reordering)
        # self.pwff = GateFFNLayer(d_model,dropout_rate=0.1)
    def forward(self, queries, keys, values,prev, attention_mask=None):
        # att = self.mhatt(queries, keys, values)
        # att = self.mhatt(queries, keys, values, attention_mask, attention_weights)
        gate_res,prev = self.attn(queries,prev) if exists(self.attn) else None

        att = self.mhatt(queries,gate_res = gate_res)

        ff = self.pwff(att)
        print(ff.shape)
        return ff,prev
```

图 2. 代码对比图，上面为原始代码，下面为改进代码

4.4 创新点

(1) 在编码端，提出了一种基于外部先验和自先验注意力的方法 EPSPA，用于解决 Transformer 编码器多头自注意力机制在生成描述过程中忽略样本间联系和缺少先验知识指导的问题。其中，在外部先验模块中，所有的输入共同更新一个权重矩阵，使得这个矩阵可以隐式地代表全部样本中最具信息量的特征。另一方面，自先验注意力模块在不增加额外内存开销的前提下将前层的注意力权重作为模拟的先验知识向后传递，进而指导模型提取到更显著的图像特征。（2）在公开数据集上对本文提出的 EPSPA 方法进行了实验，和其他方法相比，EPSPA 在参数量更低的同时在各项指标上取得了更优越的效果

5 实验结果分析

本部分首先介绍实验所用的数据集，评价指标。然后为了验证 EPSPA 方法的有效性，将所提方法与其他主流方法进行对比，最后从定性角度进行实验结果分析。

5.1 数据集与评价指标

本文使用的 MSCOCO 数据集是当前图像描述领域经常使用的大型公开数据集，该数据集包含超过 12 万张图像，其中每幅图像至少有 5 条人工标注的参考描述。实验遵循 Karpathy 等 [3] 提供的分割方法，其中 5000 张图片用于验证，5000 张图片用于测试，其余图片用于训练。

在测试阶段，本文使用了一套完整的评价指标，分别为 BLEU，METEOR，ROUGE，CIDEr，和 SPICE。其中，BLEU 是用于评估机器生成文本质量的指标；METEOR 是用于评估自动机器翻译的指标；ROUGE 基于最长公共子串来计算准确率；CIDEr 用于评测生成描述和参考描述的相似度;SPICE 是一种基于场景图和语义概念的评估指标，用于衡量生成语句是否描述了图像中各个对象之间的关系。

5.2 与先进方法对比实验

EPSPA 与当前主流图像描述生成方法在 MSCOCO 数据集上进行实验对比，其中 M2 是参考的文献，其他方法的数值也是参考 M2 论文中的实验部分。由图 3可知，EPSPA 与其他方法相比，表现出良好的性能。EPSPA 在 CIDEr 指标和 SPIC-E 指标上超过了所有其他方法，同时在 BLEU-1，METEOR 指标的表现上具有最好的竞争力。其中，它将 CIDEr 的指标提高了 0.3。这充分说明本文提出的 EPSPA 方法可以一定程度的利用样本间的潜在相关性，使模型获得良好的性能表现与泛化能力，从而提高模型生成描述的准确性。

Model	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE
SCST	-	34.2	26.7	55.7	114.0	-
Up-Down	79.8	36.3	27.7	56.9	120.1	21.4
RFNet	79.1	36.5	27.7	57.3	121.9	21.2
VRCRA	80.6	37.9	28.4	58.2	123.7	21.8
GCN-LSTM	80.5	38.2	28.5	58.3	127.6	22.0
SGAE	80.8	38.4	28.4	58.6	127.8	22.1
ORT	80.2	38.6	28.7	58.4	128.3	22.6
AoANet	80.2	38.9	29.2	**58.8**	129.8	22.4
M2	80.8	**39.1**	29.2	58.6	131.2	22.6
EPSPA	**80.8**	38.6	**29.3**	58.5	**131.5**	**22.7**

图 3. 与先进方法的实验对比

5.3 性能参数比实验

为了证明 EPSPA 方法的轻量级性能，将其与当前主流图像描述方法在参数量和描述的生成质量上进行了综合实验。由于外部先验模块不依赖复杂的注意力计算，仅仅引入一个固定大小的循环矩阵，因此外部先验模块的计算复杂度比自注意力机制要低。同时，相较于许多使用多头注意力的主流方法，本文只使用了单头自先验注意力以减少参数量。图 2 中横轴代表参数量，纵轴代表 CIDEr 指标。实验结果如图 4所示，EPSPA 方法不仅参数量在所有方法中是最低的，而且得到了最高的 CIDEr 指标

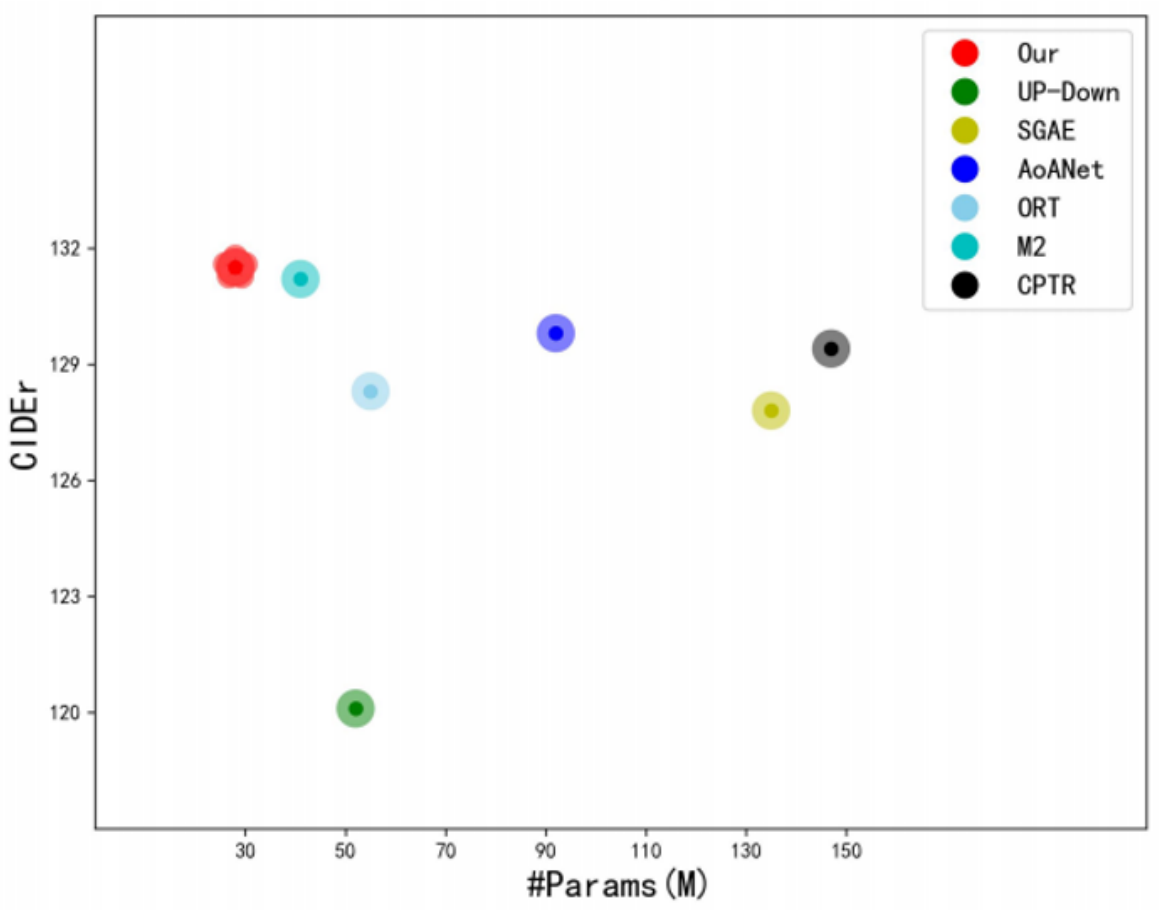

图 4. 性能-参数图

5.4 实验结果定性分析

为了对 EPSPA 性能有一个直观的理解，本文选取部分实验结果展示在图 5。其中 GT 表示参考描述。经过对比可以很直观地发现，EPSPA 可以更好的捕捉图像中对象的颜色，对象之间的空间关系以及图像细节。

例如图中（a）和（b）所示，Transformer 只能描述摩托车和消防栓的大致方位，但是 EPSPA 可以捕获样本间的潜在相关性，借助其他图片中学习到的颜色特征把摩托车和消防栓的颜色给描述出来。同时，EPSPA 使用自先验注意力可以关注到图像中更相关的区域，从而可以生成“truck”和“brick sidewalk”

例如图中（d）所示，EPSPA 可以检测出更多细节，更能把句子表达完整。EPSPA 生成的句子中不仅把“wearing”这个动作表达出来，更是可以判断出男子的表情“smiling”，这是 Transformer 无法实现的。正如上述例子所展现的，EPSPA 可以捕获图片中更详细的上下文信息，指导模型关注更相关的区域，进而生成更准确的图像描述。

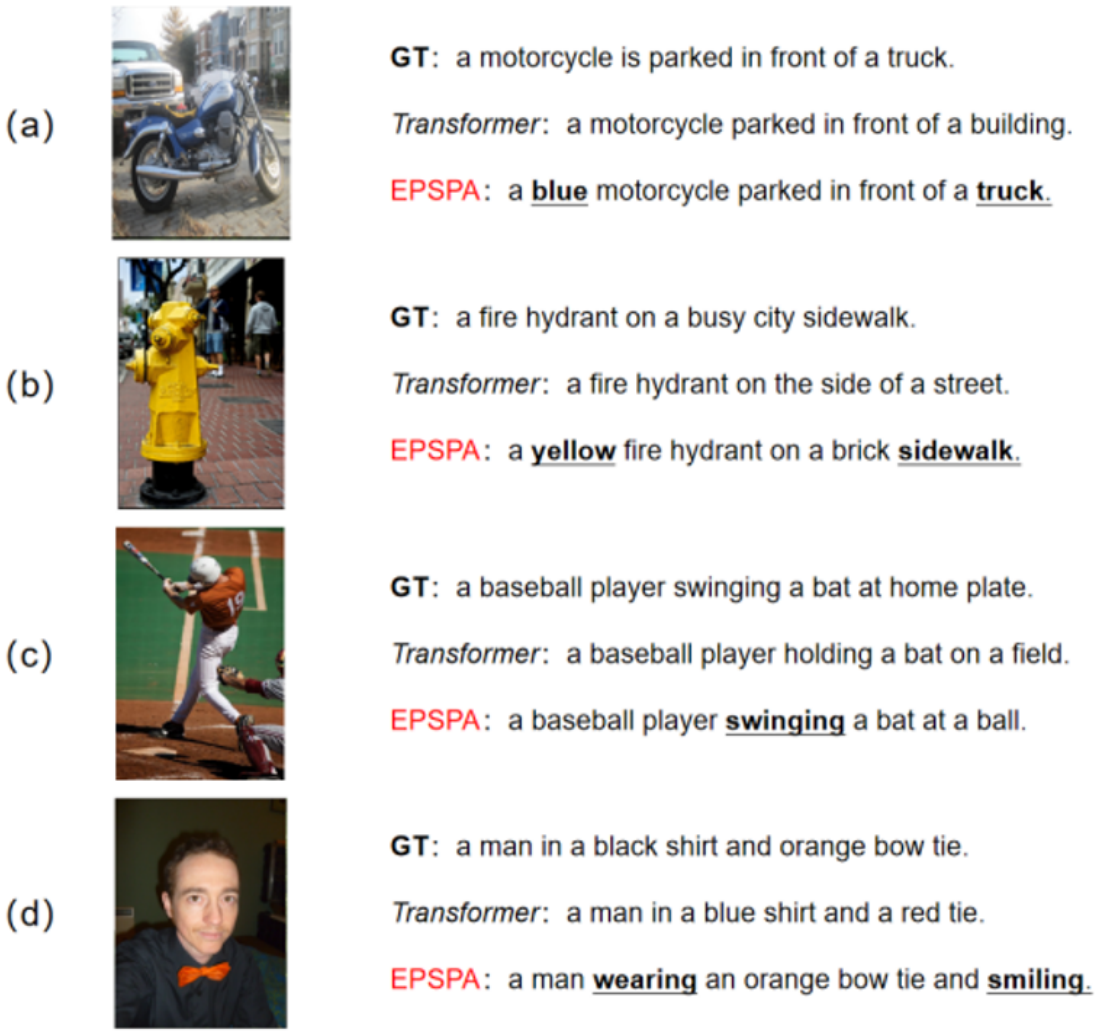

图 5. EPSPA 模型生成的图像描述

6 总结与展望

本文提出了一种基于外部先验和自先验注意力的图像描述生成方法 EPSPA。该方法考虑到了不同样本之间的潜在相关性，以及利用经过学习的注意力权重模拟先验知识，使得生成的描述更准确。首先，为了联系样本间的视觉语义，本文提出外部先验模块。它利用共享权重矩阵的优势对不同样本之间的潜在相关性进行获取。其次，在不增加额外内存的前提下，自先验注意力能够通过传递层间信息模拟先验知识，进而指导模型进行特征提取。最后，将经过融合形成的视觉特征进行解码，生成更准确的图像描述。实验结果表明，该方法在保持最低参数量的同时能够展现出超越当前主流方法的性能。在未来的工作中，可以在编码器尝试融入更多的图像特征信息，在解码端尝试对图像特征进行更深层次的交互，提出描述更为准确的轻量化图像描述方法。

参考文献

[1] I Sutskever. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

[2] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[3] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[4] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Show, control and tell: A framework for generating controllable and grounded captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8307–8316, 2019.

[5] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[6] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[7] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4634–4643, 2019.

[8] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10578–10587, 2020.

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[10] Kelvin Xu. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Minh-Thang Luong. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[13] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[14] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. *Advances in neural information processing systems*, 32, 2019.

[15] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

[16] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in neural information processing systems*, 34:9204–9215, 2021.