METransformer

1. 引入多个专家token，作为可以学习的embeddeding，在编码器中，专家token与视觉的token以及其他的专家token交互，从而学会关注图像的不同区域
2. 每个专家token在编码器中关注不同的图像区域，通过正交损失的方式实现
3. 解码的时候，专家token用于引导单词和视觉token的嵌入学习，从而生成M分候选报告，随后采用基于指标的专家投票策略选出最佳报告

编码器：从文本或图像中提取特征，生成一种嵌入向量的高维表示；Vit和专家双线性变换编码器

解码器：接受编码器的输出，用自回归的方式生成预测结果

token：原始输入的一个组成部分

embedding：将离散的token表示为连续的数值表示，每个toekn会经过一个embedding layer的操作转换为一个高维向量，这个向量就是token的embedding；

METransformer的设计目的是自动生成放射影像的诊断报告。它的总体架构分为三个主要步骤：

1. **输入阶段**：影像被切分成小块，每块成为一个"视觉Token"（Visual Token），作为编码器的输入。
2. **编码阶段**：影像的视觉Token和多个"专家Token"被输入到编码器。编码器处理这些Token，并将它们转换成包含语义信息的嵌入向量。
3. **解码阶段**：解码器使用编码器输出的嵌入向量生成报告。每个专家Token生成一个候选报告，最终通过投票选择出最佳报告作为最终输出。

医学影像报告生成可以分为两个主要研究方向：

(1) 改进模型结构

这一方向的研究主要集中在通过引入更好的注意力机制或优化解码器结构来提高模型性能。例如：

- **层次化LSTM网络**：一些研究使用层次化结构的LSTM网络来更好地生成医学报告的长文本。例如，Jing等人提出了一个多任务的层次化模型，通过预测关键词来生成长段落。
- **分段生成结构**：Xue等人提出了一种不同的网络结构，包含生成句子模型和生成段落模型，以先生成一个句子，然后利用该句子来生成下一个句子。
- **图像-报告匹配网络**：Wang等人设计了一种图像-报告匹配网络，减少了图像与文本之间的差异，使生成的报告更加准确。
- **Transformer解码器**：有些研究使用Transformer取代LSTM作为解码器，这种方法在生成报告方面效果显著。例如，一项工作提出了一种基于记忆驱动的Transformer，能够在生成过程中记录关键信息，从而提高报告的完整性。

(2) 引入医学领域知识

另一个研究方向是**利用医学领域的知识来指导报告生成**，提升报告的质量。这种方法通常包含以下几种手段：

- **知识图谱的整合**：很多最新的研究尝试将知识图谱整合到报告生成流程中，利用医学知识来改进报告的准确性和连贯性。
- **疾病标签**：另一类方法使用疾病标签来帮助生成报告。这些标签提供了疾病相关的信息，使模型更清楚地理解图像内容。
- **基于一般和特定知识的框架**：Yang等人提出了一个基于通用和特定知识的框架，通用知识来源于预构建的知识图谱，而特定知识则是通过检索相似的报告获得的。

例如，Yang等人的方法利用知识图谱提供了一个总体背景，而类似报告的检索提供了更个性化的细节，从而优化报告生成的质量。

多专家Vit编码器：

- Vit（Vision Transformer）编码器：将图像切分为patchs并转换为向量（视觉token），再利用Transformer的注意力机制处理这些视觉token提取图像特征。METransform还加入了专家Token
- 图像x HWC H高 W宽 C通道数。图像被切分为若小块，每一小块大小为P*P像素，所以一共可以被切分为*HW/p2块
- 再把每一个小块flatten成一个向量，映射到指定维度D，形成视觉Token 分块序列xp∈N*（p2*c）就视为输入序列的视觉token
- **专家token**：增加了M个专家的token，每个token的维度也是D，表示不同专家的关注区域，专家token可以训练自动调整，引入了**正交损失**，确保每个专家token关注的区域相互不同

  最后视觉token和专家token一同作为编码器的输入
- **段嵌入**：用段标记区分视觉token和专家token，每个token加上了一个对应的段嵌入Eseq
- **位置嵌入**：为视觉token和专家token添加一维可学习的位置嵌入Epos，让输入序列有序
- **模型结构**：

1. 输入初始化公式

输入初始化的公式如下：

$$z_0 = [x_p^1 E; x_p^2 E; \ldots; x_p^N E; x_e^1; x_e^2; \ldots; x_e^M] + E_{pos} + E_{seg}$$

解释：

- **视觉Token和专家Token的定义：**
  - $x_p^1, x_p^2, \ldots, x_p^N$：这是图像的视觉Token，代表分割后的图像块的嵌入。
  - $x_e^1, x_e^2, \ldots, x_e^M$：这是模型中引入的专家Token的嵌入，每个专家Token代表一位"虚拟专家"，关注影像的不同区域。
- **Token嵌入映射：**
  - 视觉Token x_p^i 被映射到一个高维向量空间，表示成 $x_p^i E$，其中 E 是一个可学习的映射矩阵。
  - 这个映射将视觉Token和专家Token都转换为相同的向量维度 D，确保后续处理的统一性。
- **位置嵌入和段嵌入：**
  - E_{pos} 是位置嵌入矩阵，为每个Token提供位置信息（即Token在序列中的位置）。
  - E_{seg} 是段嵌入矩阵，用于区分视觉Token和专家Token，以便模型理解这些Token的不同来源。

最终，z_0 表示将所有视觉Token、专家Token、位置嵌入和段嵌入相加后得到的初始输入向量。

2. 多头自注意力公式

多头自注意力层的计算公式为：

$$\hat{z}_l = \mathrm{MSA}(\mathrm{LN}(z_{l-1})) + z_{l-1}$$

解释：

- **多头自注意力（MSA）**：多头自注意力是一种注意力机制，能够在输入Token之间建立依赖关系。它的作用是计算每个Token对其他Token的关注权重。
  - 公式中的 $\mathrm{MSA}(\mathrm{LN}(z_{l-1}))$ 表示应用多头自注意力机制，其中 z_{l-1} 是第 $l-1$ 层的输出。
- **层归一化（Layer Normalization, LN）**：在计算多头注意力之前，先对输入进行层归一化，使得输入具有统一的尺度，有助于提高模型的稳定性。
- **残差连接**：$+z_{l-1}$ 表示加入残差连接。残差连接将上一层的输出 z_{l-1} 直接加到当前层的输出上，这种设计有助于信息的传递，防止深层网络出现梯度消失的问题。

结果 $\hat{z}_l$ 是经过多头注意力层和残差连接后的输出。

3. 多层感知机（MLP）层公式

MLP层的计算公式为：

$$z_l = \text{MLP}(\text{LN}(\hat{z}_l)) + \hat{z}_l$$

解释：

- **MLP层**：MLP层由全连接网络（即多层感知机）组成，用于进一步提取特征。它包含一个或多个线性层和激活函数。
- **层归一化（LN）**：在MLP层的输入上先做层归一化 $\text{LN}(\hat{z}_l)$，确保输入的均值和方差在每层之间保持一致。
- **残差连接**：与多头注意力层类似，MLP层的输出也通过残差连接，与输入直接相加，以增强网络的深度信息流通性。

结果 z_l 是经过MLP层和残差连接后的输出，它会作为下一层的输入。整个过程不断重复，直到编码器的所有层都处理完毕。

总结

这些公式描述了METransformer编码器的关键步骤：

1. **输入初始化**：将视觉Token和专家Token映射成统一的向量，并加入位置和段信息。
2. **多头自注意力**：计算每个Token之间的相关性，通过残差连接保持信息的流动。
3. **MLP层**：进一步处理特征，通过非线性变换捕捉更复杂的模式，同时保留信息流动。

多专家双线性注意力编码器：

可以将视觉token和专家token结合在一起，通过更加双线性注意力来捕捉细粒度信息。

Vit编码器的输出：

首先，METransformer的ViT编码器会输出一组包含视觉Token和专家Token的嵌入向量 $z_L \in \mathbb{R}^{(N+M)\times D}$：

- **视觉Token嵌入**：记为 $z_L^v = z_L[:N]$，表示从编码器输出中截取的前 N 个视觉Token。
- **专家Token嵌入**：记为 $z_L^e = z_L[N:(N+M)]$，表示从编码器输出中截取的后 M 个专家Token。

这些视觉Token和专家Token通过多头自注意力（即线性注意力）进行了初步的交互处理。由于医学图像的细粒度特征，模型在此基础上进一步引入**双线性注意力**机制，以增强Token之间的关联性。

双线性注意力EBA：

2. 双线性注意力（Expert Bilinear Attention, EBA）

双线性注意力（EBA）是该编码器的核心组件，用于计算视觉Token和专家Token之间的更高阶的注意力。步骤如下：

2.1 查询、键和值的定义

- 使用增强后的专家Token嵌入 z_L^e 作为**查询（Query, Q）**，即 $Q \in \mathbb{R}^{M\times 1\times D_q}$。
- 使用视觉Token嵌入 z_L^v 作为**键（Key, K）**和**值（Value, V）**：
  - $K \in \mathbb{R}^{1\times N\times D_k}$，表示对视觉Token的键向量。
  - $V \in \mathbb{R}^{1\times N\times D_v}$，表示对视觉Token的值向量。

2.2 双线性池化

为了计算专家Token和视觉Token的交互，将查询和键、值的向量进行**低秩双线性池化（low-rank bilinear pooling）**，生成**双线性查询-键**和**双线性查询-值**，公式如下：

$$B_k = \sigma(W_k K) \odot \sigma(W_{qk} Q)$$

$$B_v = \sigma(W_v V) \odot \sigma(W_{qv} Q)$$

其中：

- $W_k \in \mathbb{R}^{D_B\times D_k}$、$W_{qk} \in \mathbb{R}^{D_B\times D_q}$ 是键相关的可学习参数。
- $W_v \in \mathbb{R}^{D_B\times D_v}$、$W_{qv} \in \mathbb{R}^{D_B\times D_q}$ 是值相关的可学习参数。
- σ 表示ReLU激活函数，$\odot$ 表示逐元素乘积（即Hadamard积）。

计算得到的 $B_k \in \mathbb{R}^{M\times N\times D_B}$ 和 $B_v \in \mathbb{R}^{M\times N\times D_B}$ 分别是双线性查询-键和双线性查询-值。

3. 空间和通道的注意力计算

为了进一步细化注意力机制，模型在**空间维度**和**通道维度**上分别计算注意力权重：

3.1 空间注意力（Spatial-wise Attention）

- 通过一个线性层将 B_k 投影到一个中间表示 B_{mid}：

  $$B_{\text{mid}} = \sigma(W_{B_k} B_k)$$

  其中 $W_{B_k} \in \mathbb{R}^{D_B\times D_{\text{mid}}}$ 是可学习参数。
- 然后再通过另一个线性层将 B_{mid} 从维度 D_{mid} 映射到 1，之后应用Softmax获得空间注意力权重：

  $$\alpha_s = \text{softmax}(B_{\text{mid}})$$

  最终得到的 $\alpha_s \in \mathbb{R}^{M\times N\times 1}$ 是空间维度的注意力权重。

3.2 通道注意力（Channel-wise Attention）

- 使用**Squeeze-and-Excitation**操作（即平均池化），对 B_{mid} 进行通道注意力计算：

  $$\beta_c = \text{sigmoid}(W_c \bar{B}_{\text{mid}})$$

  其中：
  - $W_c \in \mathbb{R}^{D_{\text{mid}}\times D_B}$ 是可学习的参数矩阵。
  - $\bar{B}_{\text{mid}} \in \mathbb{R}^{M\times D_{\text{mid}}}$ 表示 B_{mid} 在 N 维度上的平均池化。

得到的 $\beta_c \in \mathbb{R}^{M\times D_B}$ 是通道维度上的注意力权重。

3.3 综合空间和通道注意力

综合得到的空间和通道注意力权重，计算双线性注意力的输出：

$$\hat{z}_L^{e(1)} = \text{EBA}(\hat{z}_L^e, \hat{z}_L^v) = \beta_c \odot \alpha_s B_v$$

即，输出的专家嵌入 $\hat{z}_L^{e(1)}$ 是经过空间和通道注意力加权的值向量 B_v。

4. 双线性编码器层

双线性编码器层通过在标准Transformer编码器中加入EBA和残差连接来进一步处理Token嵌入：

- 第 n 层**双线性编码器**的计算公式如下：

  $$\hat{z}_L^{e(n)} = \text{EBA}(\hat{z}_L^{e(n-1)}, \hat{z}_L^{v(n-1)})$$

  $$\hat{z}_L^{v(n)} = \text{LN}(W_e^n [\hat{z}_L^{e(n-1)}; \hat{z}_L^{v(n-1)}] + \hat{z}_L^{v(n-1)})$$

其中：

- $W_e^n \in \mathbb{R}^{(D_q+D_v)\times D_v}$ 是可学习的线性变换矩阵。
- LN 表示层归一化操作。

这个双线性编码器层反复堆叠 N 次，每次都对视觉和专家Token之间的高阶交互进行建模，直到模型捕获到足够细粒度的医学图像信息。

4. 双线性编码器层

双线性编码器层通过在标准Transformer编码器中加入EBA和残差连接来进一步处理Token嵌入：

- 第 n 层双线性编码器的计算公式如下：

$$\hat{z}_L^{e(n)} = \mathrm{EBA}(\hat{z}_L^{e(n-1)}, \hat{z}_L^{v(n-1)})$$

$$\hat{z}_L^{v(n)} = \mathrm{LN}(W_e^n[\hat{z}_L^{e(n-1)}; \hat{z}_L^{v(n-1)}] + \hat{z}_L^{v(n-1)})$$

其中：

- $W_e^n \in \mathbb{R}^{(D_q+D_v)\times D_v}$ 是可学习的线性变换矩阵。
- LN 表示层归一化操作。

这个双线性编码器层反复堆叠 N 次，每次都对视觉和专家Token之间的高阶交互进行建模，直到模型捕获到足够细粒度的医学图像信息。

多头注意力基于查询Q和键K向量进行点积，只能捕捉查询和键向量之间的线性关系

双线性注意力通过逐元素乘积Hadamard积捕捉查询和键之间的非线性交互关系

2.1 查询、键和值的定义

在双线性注意力中，我们将查询、键和值分别定义如下：

- **查询 Q**：用专家Token的嵌入表示，形状为 $Q \in \mathbb{R}^{M\times 1\times D_q}$，其中 M 是专家Token的数量。
- **键 K 和 值 V**：用视觉Token的嵌入表示，形状为 $K \in \mathbb{R}^{1\times N\times D_k}$ 和 $V \in \mathbb{R}^{1\times N\times D_v}$，其中 N 是视觉Token的数量。

2.2 双线性池化

双线性池化用于对查询和键、值向量进行逐元素交互：

1. **查询-键双线性交互**：
   - 为了表示查询 Q 和键 K 之间的非线性关系，先将 K 和 Q 分别映射到一个共同的低维空间（维度为 D_B），再进行逐元素相乘：

$$B_k = \sigma(W_k K) \odot \sigma(W_{qk} Q)$$

   - 这里 $W_k \in \mathbb{R}^{D_B\times D_k}$ 和 $W_{qk} \in \mathbb{R}^{D_B\times D_q}$ 是两个映射矩阵，σ 是ReLU激活函数，$\odot$ 是逐元素乘积（Hadamard积）。
   - 得到的 $B_k \in \mathbb{R}^{M\times N\times D_B}$ 是一个捕捉到查询和键之间非线性关系的特征张量。
2. **查询-值双线性交互**：
   - 类似地，对查询 Q 和值 V 进行双线性交互：

$$B_v = \sigma(W_v V) \odot \sigma(W_{qv} Q)$$

   - 这里 $W_v \in \mathbb{R}^{D_B\times D_v}$ 和 $W_{qv} \in \mathbb{R}^{D_B\times D_q}$ 是值相关的映射矩阵。
   - 得到的 $B_v \in \mathbb{R}^{M\times N\times D_B}$ 是查询和值之间的双线性特征张量。

双线性池化的结果 B_k 和 B_v 包含了查询和键、值之间的复杂关系，使得模型能够捕捉到视觉和专家Token之间更高阶的依赖。

2. 双线性注意力中Q、K和V的具体生成方式

在双线性注意力模块中，我们需要从视觉Token和专家Token生成新的Q、K和V，步骤如下：

1. **Q（查询）**：
   - 将专家Token的嵌入向量 z_L^e 通过一个线性映射矩阵 W^Q 投影到查询空间：

$$Q = z_L^e W^Q$$

   - $Q \in \mathbb{R}^{M\times 1\times D_q}$，其中 M 是专家Token的数量，D_q 是查询的维度。
   - 这个过程让每个专家Token成为一个查询向量，用于表达该"专家"对视觉Token的关注需求。
2. **K（键）和V（值）**：
   - 将视觉Token的嵌入 z_L^v 分别通过线性映射矩阵 W^K 和 W^V 投影，生成键和值：

$$K = z_L^v W^K, \quad V = z_L^v W^V$$

   - $K \in \mathbb{R}^{1\times N\times D_k}$ 和 $V \in \mathbb{R}^{1\times N\times D_v}$，其中 N 是视觉Token的数量，D_k 和 D_v 分别是键和值的维度。

2.2 双线性池化

双线性池化用于对查询和键、值向量进行逐元素交互：

1. **查询-键双线性交互**：
   - 为了表示查询 Q 和键 K 之间的非线性关系，先将 K 和 Q 分别映射到一个共同的低维空间（维度为 D_B），再进行逐元素相乘：

$$B_k = \sigma(W_k K) \odot \sigma(W_{qk} Q)$$

   - 这里 $W_k \in \mathbb{R}^{D_B \times D_k}$ 和 $W_{qk} \in \mathbb{R}^{D_B \times D_q}$ 是两个映射矩阵，σ 是ReLU激活函数，$\odot$ 是逐元素乘积（Hadamard积）。
   - 得到的 $B_k \in \mathbb{R}^{M \times N \times D_B}$ 是一个捕捉到查询和键之间非线性关系的特征张量。
2. **查询-值双线性交互**：
   - 类似地，对查询 Q 和值 V 进行双线性交互：

$$B_v = \sigma(W_v V) \odot \sigma(W_{qv} Q)$$

   - 这里 $W_v \in \mathbb{R}^{D_B \times D_v}$ 和 $W_{qv} \in \mathbb{R}^{D_B \times D_q}$ 是值相关的映射矩阵。
   - 得到的 $B_v \in \mathbb{R}^{M \times N \times D_B}$ 是查询和值之间的双线性特征张量。

双线性池化的结果 B_k 和 B_v 包含了查询和键、值之间的复杂关系，使得模型能够捕捉到视觉和专家Token之间更高阶的依赖。

3. 空间和通道注意力的计算

双线性池化得到的 B_k 和 B_v 中的信息量较大，因此通过进一步的空间和通道注意力机制来解读它们的特征。

3.1 空间注意力

空间注意力的目的是让模型学会在空间上关注不同位置的Token：

- 先通过一个线性层将 B_k 转换到中间特征空间 B_{mid}：

$$B_{\text{mid}} = \sigma(W_{B_k} B_k)$$

- 然后通过另一线性层将 B_{mid} 映射到维度为1，最终通过Softmax函数得到空间维度的注意力权重 α_s：

$$\alpha_s = \text{softmax}(B_{\text{mid}})$$

- 结果 $\alpha_s \in \mathbb{R}^{M \times N \times 1}$ 表示每个专家Token在不同视觉Token位置上的注意力权重。

3.2 通道注意力

通道注意力的目的是在特征通道上对重要信息进行加权：

- 首先对 B_{mid} 进行平均池化，得到 $\bar{B}_{\text{mid}}$：

$$\bar{B}_{\text{mid}} = \frac{1}{N} \sum_{i=1}^{N} B_{\text{mid}}[i]$$

- 然后通过一个全连接层和激活函数（sigmoid），得到通道注意力权重 β_c：

$$\beta_c = \text{sigmoid}(W_c \bar{B}_{\text{mid}})$$

- 结果 $\beta_c \in \mathbb{R}^{M \times D_B}$ 表示每个通道的注意力权重。

4. 最终输出

通过结合空间和通道注意力权重，最终的双线性注意力结果为：

$$\hat{z}_L^{e(1)} = \text{EBA}(\hat{z}_L^e, \hat{z}_L^v) = \beta_c \odot \alpha_s B_v$$

这个公式表示对值向量 B_v 进行空间和通道注意力的加权求和，使得最终的专家Token嵌入包含了对细粒度特征的高阶交互信息。

总结

双线性注意力机制的设计源于捕捉更高阶的依赖关系，通过双线性池化、空间和通道注意力，使得模型能灵活地在复杂图像结构中定位和强调关键特征。

引入了残差连接（Residual Connection）。这些设计结合在一起，提升了双线性编码器的性能和稳定性。以下是对每个公式和结构的详细解释。

1. 双线性编码器的输入与输出定义

该双线性编码器层的输入包括上一层的专家Token嵌入和视觉Token嵌入：

- **初始输入**：在双线性编码器的第一层，专家Token和视觉Token的初始输入分别为 $\hat{z}_L^{e(0)} = z_L^e$ 和 $\hat{z}_L^{v(0)} = z_L^v$，即来自ViT编码器的输出。

2. 双线性注意力模块（EBA）的应用

在第 n 层双线性编码器中，首先将上一层的专家Token和视觉Token嵌入输入到**EBA模块**，计算第 n 层的专家Token嵌入：

$$\hat{z}_L^{e(n)} = \mathrm{EBA}(\hat{z}_L^{e(n-1)}, \hat{z}_L^{v(n-1)})$$

其中：

- $\mathrm{EBA}(\cdot)$ 表示专家双线性注意力模块，它对输入的专家Token嵌入和视觉Token嵌入进行交互计算。
- 输出的 $\hat{z}_L^{e(n)}$ 是更新后的专家Token嵌入，经过EBA模块的高阶特征交互后，每个专家Token都包含了丰富的图像区域信息。

3. 加法与归一化（Add & Norm）和残差连接

为了提升模型的稳定性和训练效果，双线性编码器层引入了"加法与归一化"层，同时加入了残差连接。这一部分的计算如下：

$$\hat{z}_L^{v(n)} = \mathrm{LN}(W_e^n[\hat{z}_L^{e(n-1)}; \hat{z}_L^{v(n-1)}] + \hat{z}_L^{v(n-1)})$$

其中：

- **输入拼接**：这里将前一层的**专家Token嵌入** $\hat{z}_L^{e(n-1)}$ 和**视觉Token嵌入** $\hat{z}_L^{v(n-1)}$ 拼接在一起，表示成 $[\hat{z}_L^{e(n-1)}; \hat{z}_L^{v(n-1)}]$。这种拼接方式将专家信息和视觉信息融合，以增强视觉Token对专家信息的感知。
- **线性变换**：拼接后的嵌入通过一个线性变换 W_e^n，映射到视觉Token的嵌入维度 D_v。这个线性层的作用是将拼接后的信息组合成视觉Token可以处理的维度。
- **残差连接**：$+\hat{z}_L^{v(n-1)}$ 表示加入残差连接，即将线性变换的结果与前一层的视觉Token嵌入直接相加。残差连接的作用是防止梯度消失，并帮助保留视觉Token的原始信息。
- **层归一化（Layer Normalization, LN）**：在相加之后对结果进行归一化，层归一化可以稳定训练过程，确保不同层次的嵌入特征具有均衡的尺度。

最终，得到更新后的视觉Token嵌入 $\hat{z}_L^{v(n)}$，它既包含了原始的视觉Token信息，又结合了专家Token带来的高级特征交互。

架构图：

从这个图和架构来看，该模型架构主要包含以下三个模块：**专家ViT编码器**、**双线性编码器**和**专家Transformer解码器**。每个模块的功能如下：

1. Vision Transformer (ViT) Encoder

- **输入处理**：图像被分割成若干个小的图像补丁（patches），并通过线性投影变换为视觉token嵌入。这些补丁成为视觉token，通过Vision Transformer编码器进行特征提取。
- **专家token嵌入**：除了视觉token，还引入了M个"专家token"，这些token在整个过程中具有特定的"专家"功能，用于捕捉图像的某些关键特征。
- **编码过程**：图像经过ViT编码器的多层自注意力层和MLP层的处理，将视觉token和专家token编码为高维特征表示。ViT编码器主要处理图像的**一阶交互**，即视觉token和专家token之间的初步信息交换。

2. Expert Bilinear Encoder

- **目的**：由于医学图像的细粒度特征，ViT编码器的一阶交互可能不足以充分捕获这些特征。专家双线性编码器（Expert Bilinear Encoder）进一步增强了视觉token和专家token的**高阶特征交互**。
- **双线性注意力机制**：
  - 在专家双线性编码器层，专家token嵌入被用作查询 Q，视觉token嵌入被用作键 K 和值 V。
  - **双线性池化**通过低秩双线性池化操作将查询和键、查询和值分别结合在一起，生成联合的查询-键表示 B_k 和查询-值表示 B_v。
  - **空间注意力和通道注意力**：在 B_k 和 B_v 之上，计算空间注意力和通道注意力。空间注意力捕捉到图像中每个区域的关系，通道注意力则帮助增强不同通道上的特征。
  - **结果**：双线性注意力的输出 $\hat{z}_L^{e(1)}$ 被用于更新专家token嵌入，进一步提高了它们的特征表达能力。

3. Expert Transformer Decoder

- **输入**：双线性编码器输出的视觉和专家token被传入解码器，作为解码过程的上下文信息。
- **解码过程**：
  - 解码器首先对目标文本（例如，报告文本）进行"右移"，即在每个时间步预测下一个词。
  - **交叉双线性注意力**（Cross-modal Bilinear Attention）：解码器引入了交叉双线性注意力机制，使得文本token能够与视觉和专家token进行高阶交互，从而生成更加细粒度和准确的报告描述。
  - **多层结构**：解码器包含多层"加法与归一化"层（Add & Norm）和"专家双线性注意力"层。交叉双线性注意力允许生成的报告逐层地融合图像中不同区域的信息。
- **输出**：通过最后的线性投影和softmax层，解码器输出每个词的概率分布，最终生成完整的目标报告文本。

总结

整个流程的关键在于多层高阶交互：

- **一阶交互**（Vision Transformer Encoder）：在ViT编码器中，通过多头自注意力机制，视觉和专家token初步交互，生成图像的高维特征表示。
- **二阶交互**（Expert Bilinear Encoder）：双线性编码器进一步增强视觉token和专家token的关系，通过双线性注意力机制来捕捉高阶特征。
- **跨模态双线性交互**（Expert Transformer Decoder）：在生成报告时，文本和图像token的高阶交互使生成的文本更具细粒度的描述能力。

该架构设计可以用于处理复杂的医学影像描述任务，通过多层高阶交互的设计，提升了模型对医学图像细节的捕捉和描述能力。

解码器：

解码器的输入来自最后一层的`Bilinear Attention Encoder`：包含两类嵌入向量

- expert token嵌入：每个在图像上关注不同的区域
- 视觉token嵌入：表示视觉信息

  被统一表示为fe和fv，作为解码器生成报告的输入

调整块：

Fadjust通过专家嵌入fe来调节视觉嵌入fv。从而为每个专家生成不同的报告

2. 调整块（Adjust Block）

为了在报告生成过程中充分利用专家令牌，调整块 F_{adjust} 通过专家嵌入 f_e 来调节视觉嵌入 f_v，从而为每个专家生成不同的报告：

$$f_v = F_{\text{adjust}}(f_e, f_v) = \sigma(W_e f_e) \odot \sigma(W_v f_v)$$

其中：

- W_e 和 W_v 是可学习参数；
- σ 是 ReLU 激活函数；
- $\odot$ 表示 Hadamard 积，用于逐元素相乘。

由于专家令牌被设计为正交，这意味着每个专家可以独立关注图像的不同区域，生成风格或内容不同的报告。

双线性解码层：

解码主要包括两个主要步骤的EBA操作：

- EBAmask：对词嵌入应用掩码注意力，捕捉上下文的依赖关系
- EBAcross：计算词嵌入和视觉嵌入之间的跨模态注意力，以将视觉信息融入词汇生成过程
- 并引入残差连接和层归一化

数学表达

第 i 层双线性解码器层的数学表达为：

1. **词嵌入掩码注意力**（Masked Attention）：

$$E_{\text{mid}}^{(i)} = \text{LN}(\text{EBAmask}(E_r^{(i-1)}) + E_r^{(i-1)})$$

其中 $E_r^{(i-1)}$ 是第 $i-1$ 层的词嵌入。

2. **跨模态注意力**（Cross-modal Attention）：

$$E_c^{(i)} = \text{LN}(\text{EBAcross}(E_{\text{mid}}^{(i)}, f_v) + E_{\text{mid}}^{(i)})$$

3. **调整并生成**（Adjust and Generate）：

$$E_r^{(i)} = \text{LN}(W_d^{(i)}[E_r^{(i-1)}; E_c^{(i)}] + E_r^{(i-1)})$$

其中 $W_d^{(i)}$ 是可学习参数，$[;]$ 表示向量连接。

4. 最终输出

通过多层的 Bilinear Decoder Layer 计算，最终的输出 $E_c^I \in \mathbb{R}^{M \times T \times D_r}$ 是并行生成的 M 个专家报告。然后对 E_c^I 进行线性映射和 softmax 操作，以预测每个词的生成概率。