咕泡科技
GUPAO TECH

多模态大模型MLLM
ViT CLIP BLIP

咕泡科技
GUPAO TECH

Vision Transformer (ViT)

AN IMAGE IS WORTH 16X16 WORDSTRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Transformers在视觉应用中的难点

在nlp中，输入transformer中的是一个序列，而在视觉领域，需要考虑如何将一个2d图片转化为一个1d的序列，最直观的想法将图片中的像素点输入到transformer中，模型训练中图片的大小是224*224=50176，而正常的bert的序列长度是512，复杂度太高

输入序列长度的改进

1）使用网络中间的特征图

用res50最后一个stage res4 的feature map size只有14*14=196，序列长度是满足预期的

2）孤立自注意力

使用local window而不是整张图，输入的序列长度可以由windows size来控制

3）轴自注意力

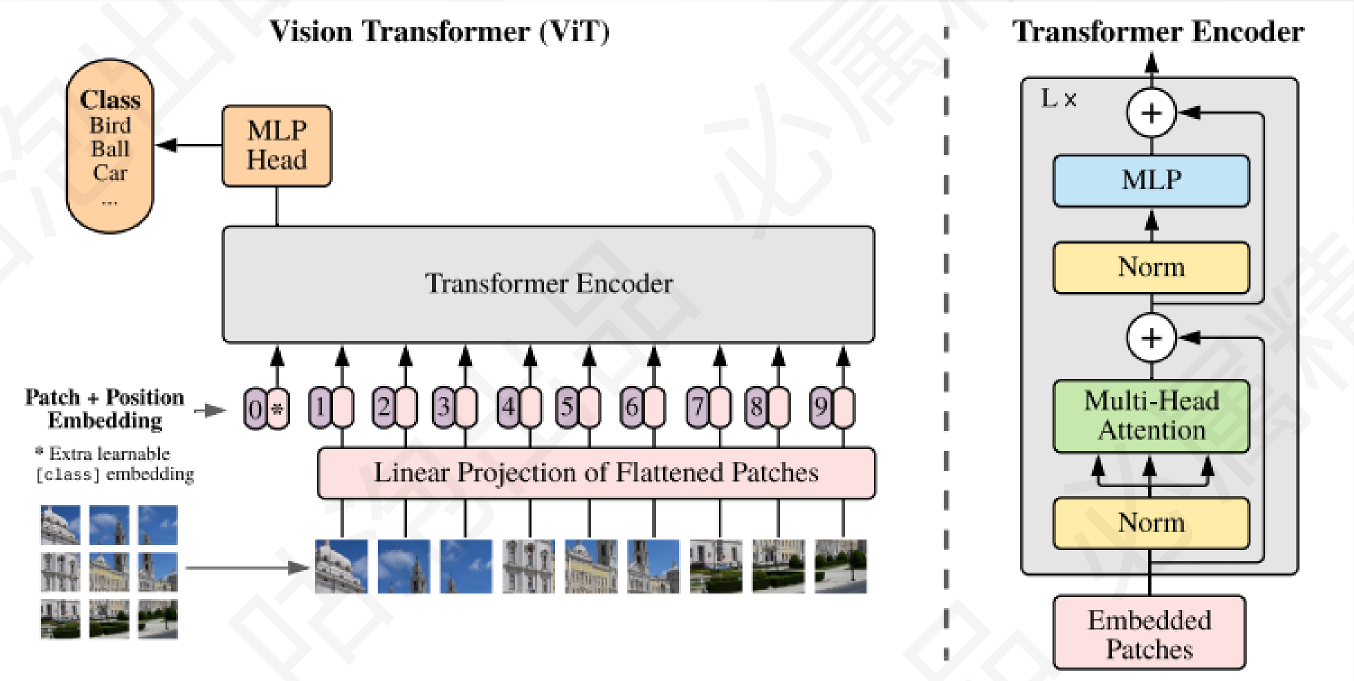
将在2d图片上的自注意力操作改为分别在图片的高和宽两个维度上做self-attention，可以大大降低复杂度，但是由于目前硬件没有对这种操作做加速，很难支持大规模的数据量级。

Vision Transformer (ViT)

AN IMAGE IS WORTH 16X16 WORDSTRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

核心结论:当拥有足够多的数据进行预训练的时候，ViT的表现就会超过CNN，突破transformer缺少归纳偏置的限制，可以在下游任务中获得较好的迁移效果

ViT的结构



ViT将输入图片分为多个patch（16x16），再将每个patch投影为固定长度的向量送入Transformer，后续encoder的操作和原始Transformer中完全相同。如果为图片分类任务，在输入序列中加入一个特殊的token，该token对应的输出即为最后的类别预测

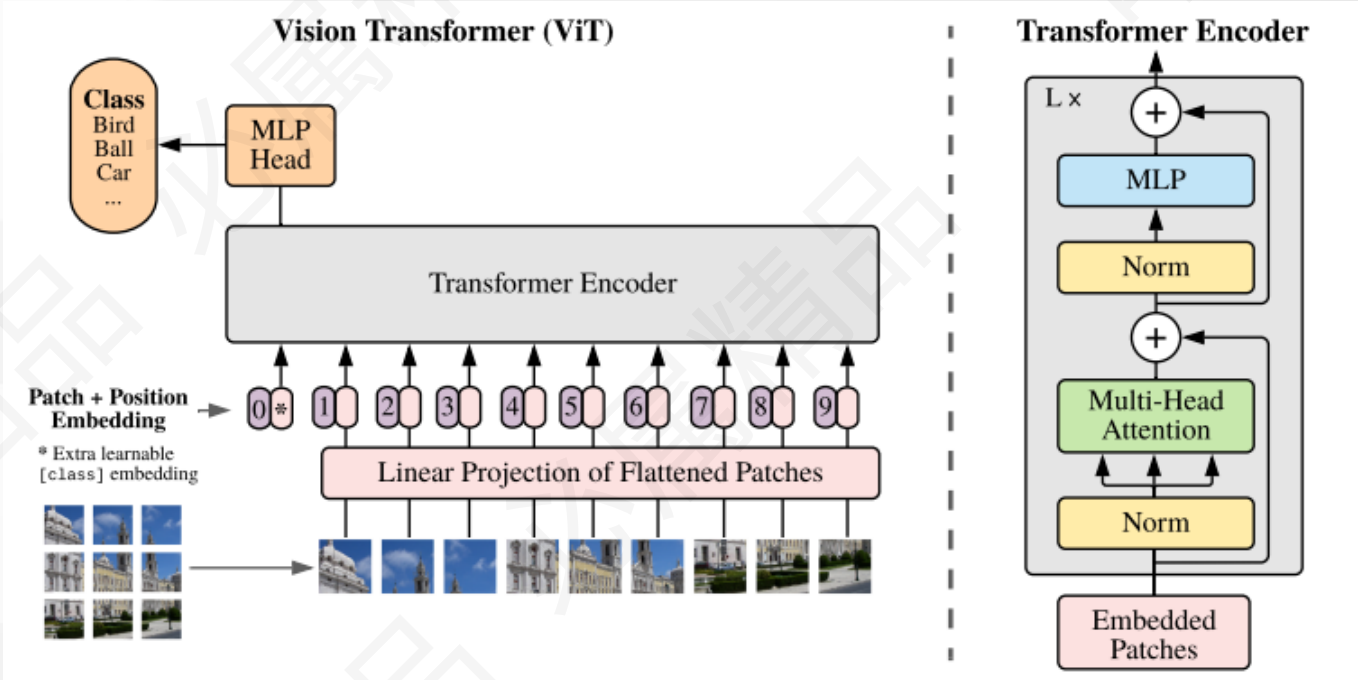
咕泡科技 GUPAO TECH

Vision Transformer (ViT)

AN IMAGE IS WORTH 16X16 WORDSTRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

VIT输入实现流程

(1) patch embedding：例如输入图片大小为224x224，将图片分为固定大小的patch，patch大小为16x16，则每张图像会生成224x224/16x16=196个patch，即输入序列长度为**196**，每个patch维度16x16x3=**768**，线性投射层的维度为768xN (N=768)，因此输入通过线性投射层之后的维度依然为196x768，即一共有196个token，每个token的维度是768。这里还需要加上一个特殊字符cls，因此最终的维度是**197x768**。到目前为止，已经通过patch embedding将一个视觉问题转化为了一个seq2seq问题

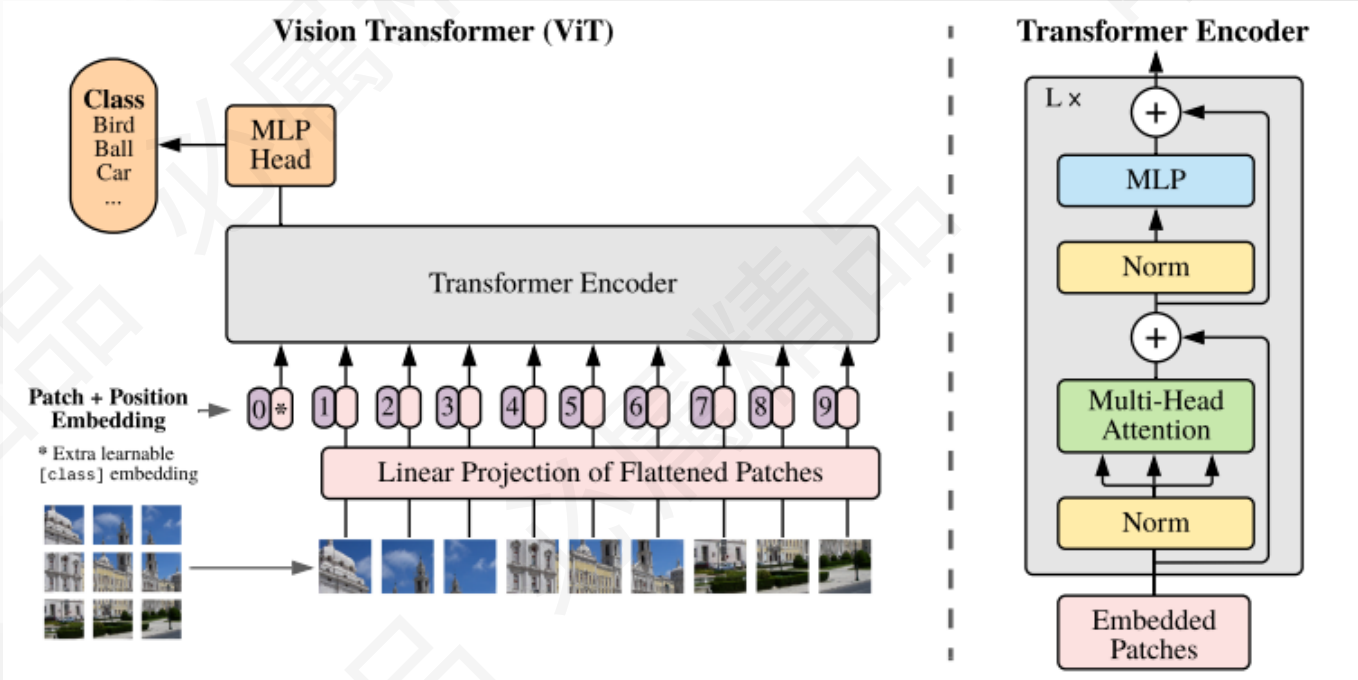


Vision Transformer (ViT)

AN IMAGE IS WORTH 16X16 WORDSTRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

VIT编码实现流程

(2) positional encoding（standard learnable 1D position embeddings）：ViT同样需要加入位置编码，位置编码可以理解为一张表，表一共有N行，N的大小和输入序列长度相同，每一行代表一个向量，向量的维度和输入序列embedding的维度相同（768）。注意位置编码的操作是sum，而不是concat。加入位置编码信息之后，维度依然是**197x768**



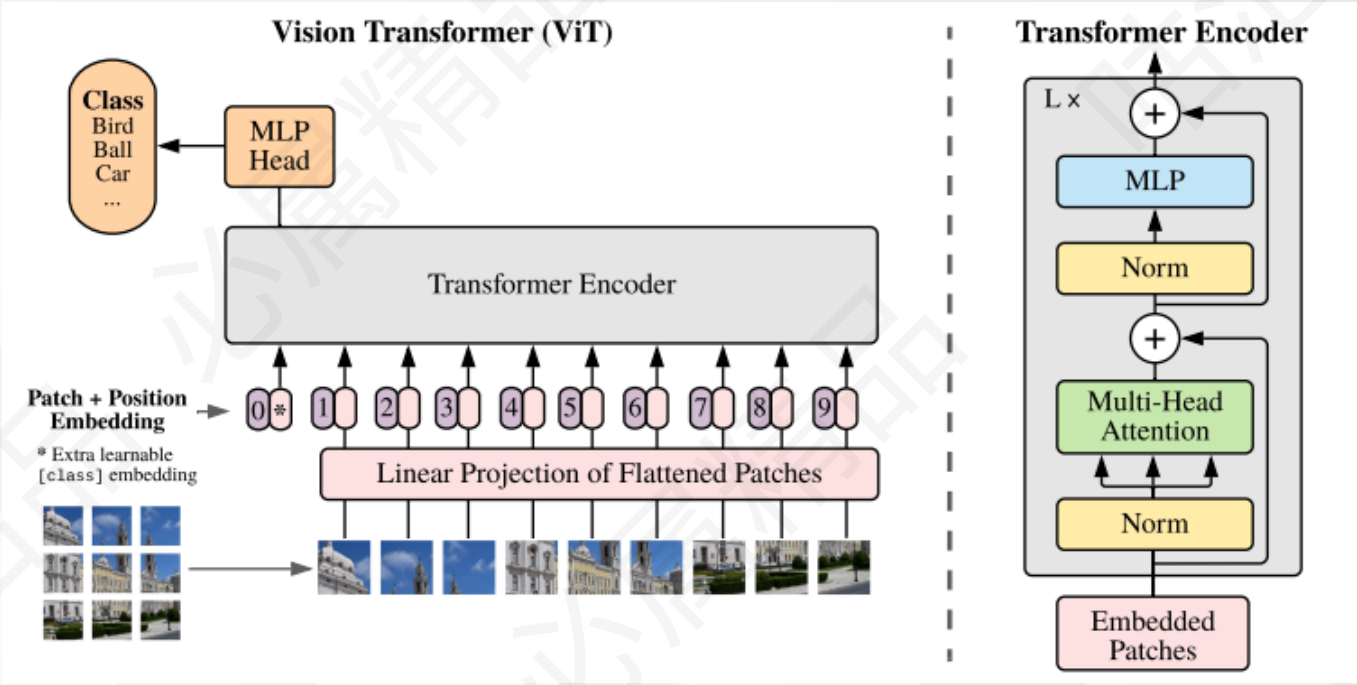
咕泡科技 GUPAO TECH

Vision Transformer (ViT)

AN IMAGE IS WORTH 16X16 WORDSTRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

VIT编码实现流程

(3) LN/multi-head attention/LN：LN输出维度依然是197x768。多头自注意力时，先将输入映射到q，k，v，如果只有一个头，qkv的维度都是197x768，如果有12个头（768/12=64），则qkv的维度是197x64，一共有12组qkv，最后再将12组qkv的输出拼接起来，输出维度是197x768，然后在过一层LN，维度依然是**197x768**



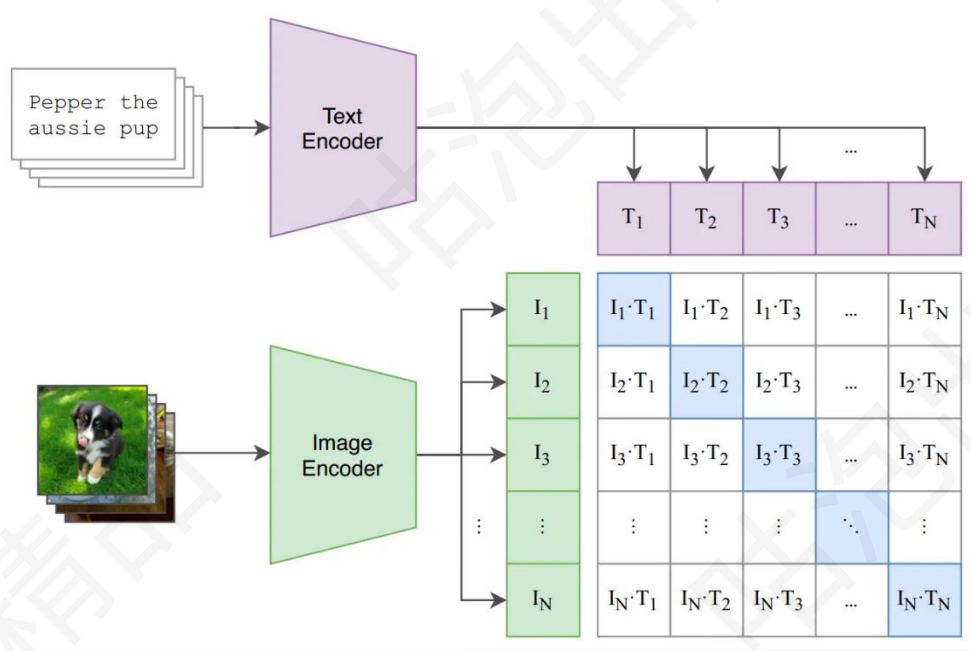
•**缺点1**：如果出现了一张图，其中包含模型从来没见过的类别，那么模型就不能输出正确的结果

•**缺点2**：如果输入数据出现了分布偏移（distribution shift），那么模型可能也无法输出正确的结果

Contrastive Language–Image Pre-training(CLIP)

CLIP预训练方法：对比学习

CLIP模型由两个主体部分组成：Text Encoder和Image Encoder——文本和图像的特征提取器

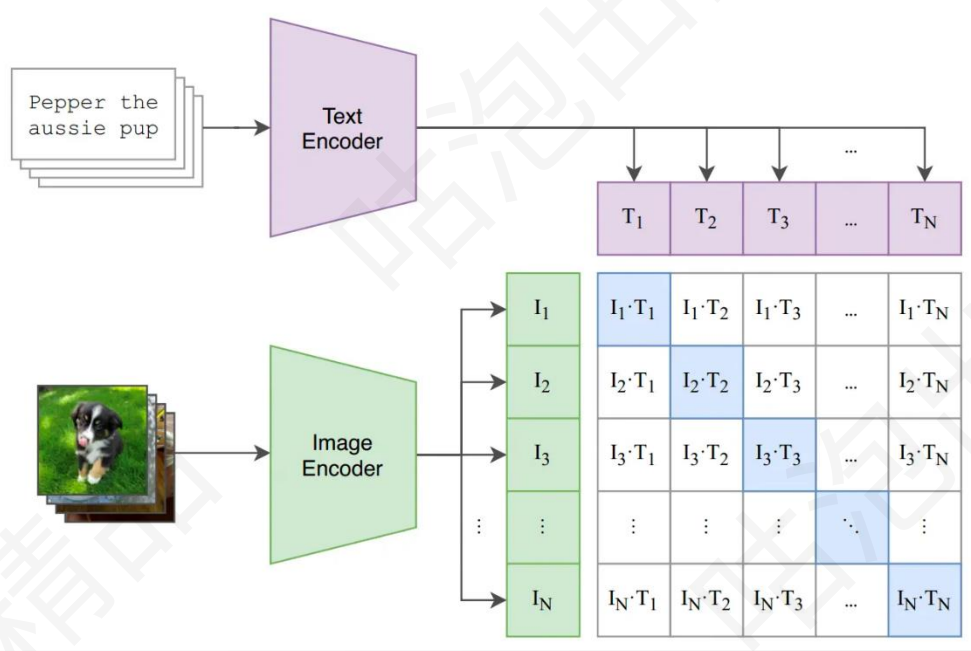


对于Text Encoder，CLIP借鉴的是GPT2架构。对于每条prompt，在进入Text Encoder前，都会添加表示开始和结束的符号[SOS]与[EOS]。最终将最后一层[EOS]位置的向量作为该prompt的特征表示向量，对于Image Encoder，CLIP则尝试过5种不同的ResNet架构和3种VIT架构，最终选用的是“ViT-L/14@336px”这个模型，也就是架构为Large，patch_size = 14的ViT，同时在整个CLIP预训练结束后，用更高分辨率（336*336）的图片做了一个epoch的fine-tune，目的是让CLIP能涌现出更好的效果。与Text Encoder类似，每张图片对应一个最终特征表示向量

Contrastive Language–Image Pre-training(CLIP)

CLIP预训练方法：对比学习

CLIP模型由两个主体部分组成：Text Encoder和Image Encoder——文本和图像的特征提取器

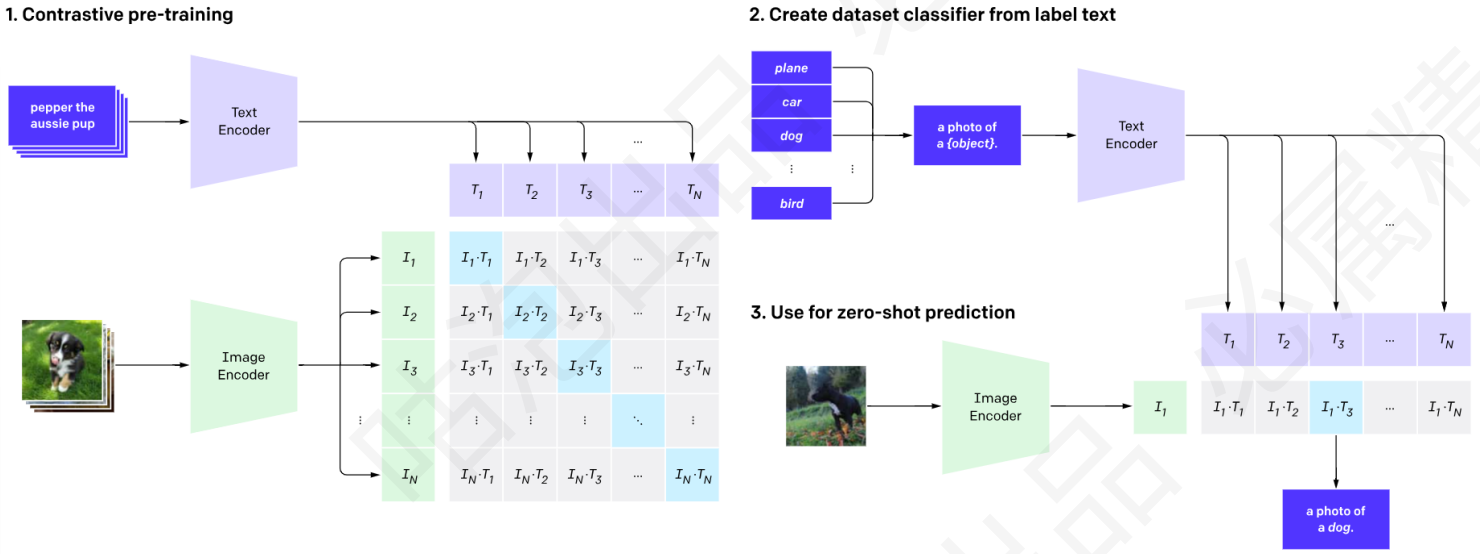


对于Text Encoder，CLIP借鉴的是GPT2架构。对于每条prompt，在进入Text Encoder前，都会添加表示开始和结束的符号[SOS]与[EOS]。最终将最后一层[EOS]位置的向量作为该prompt的特征表示向量

对于Image Encoder，CLIP则尝试过5种不同的ResNet架构和3种VIT架构，最终选用的是 “ViT-L/14@336px”这个模型，也就是架构为Large，patch_size = 14的ViT，同时在整个CLIP预训练结束后，用更高分辨率（336*336）的图片做了一个epoch的fine-tune，目的是让CLIP能涌现出更好的效果。与Text Encoder类似，每张图片对应一个最终特征表示向量

Contrastive Language–Image Pre-training(CLIP)

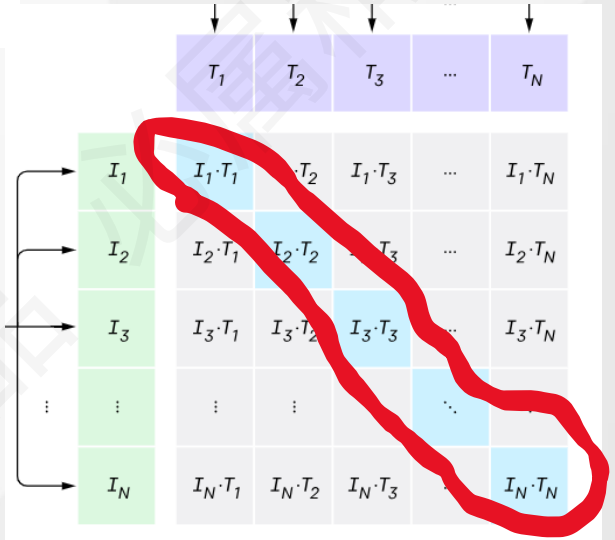
网络整体结构



- Contrastive pre-training：预训练阶段，使用图片 - 文本对进行对比学习训练；
- Create dataset classifier from label text：提取预测类别文本特征；
- Use for zero-shot predictiion：进行 Zero-Shot 推理预测；

预训练阶段

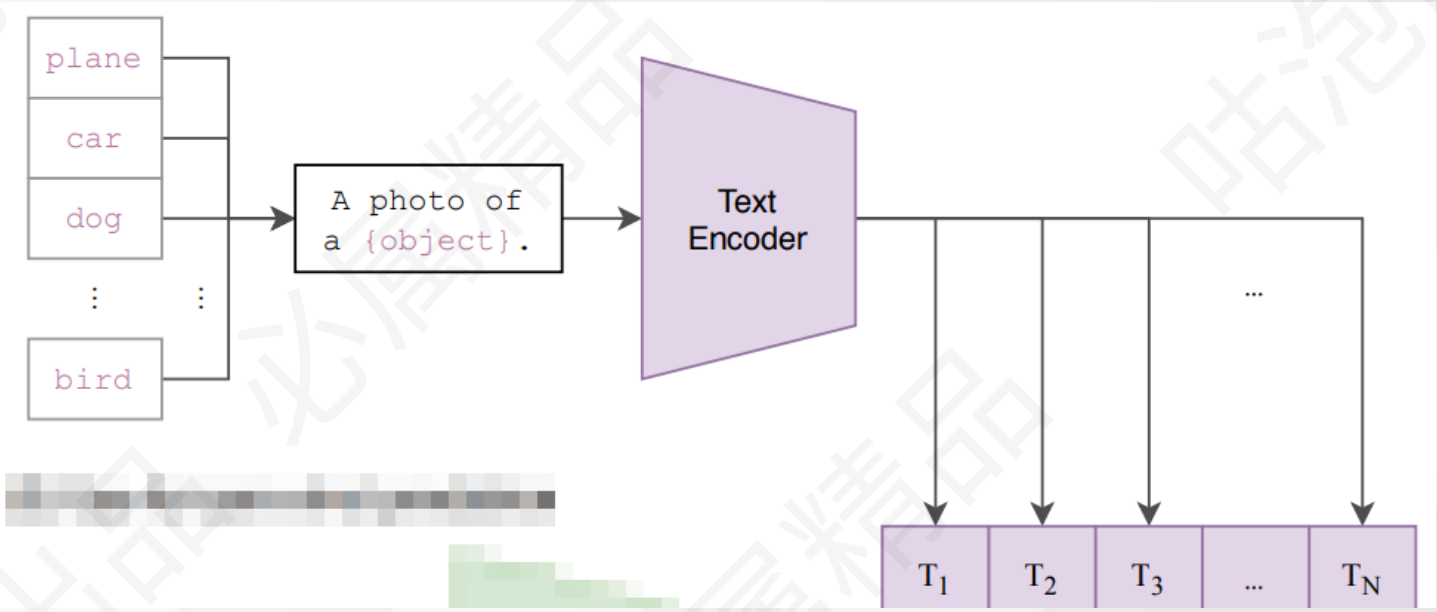
在预训练阶段，对比学习中正样本对和负样本的定义为能够配对的图片-文本对，和不能匹配的图片-文本对。具体来说，先分别对图像和文本提特征，这时图像对应生成 I1、I2 ... In 的特征向量，文本对应生成 T1、T2 ... Tn 的特征向量，然后中间对角线为正样本，其余均为负样本



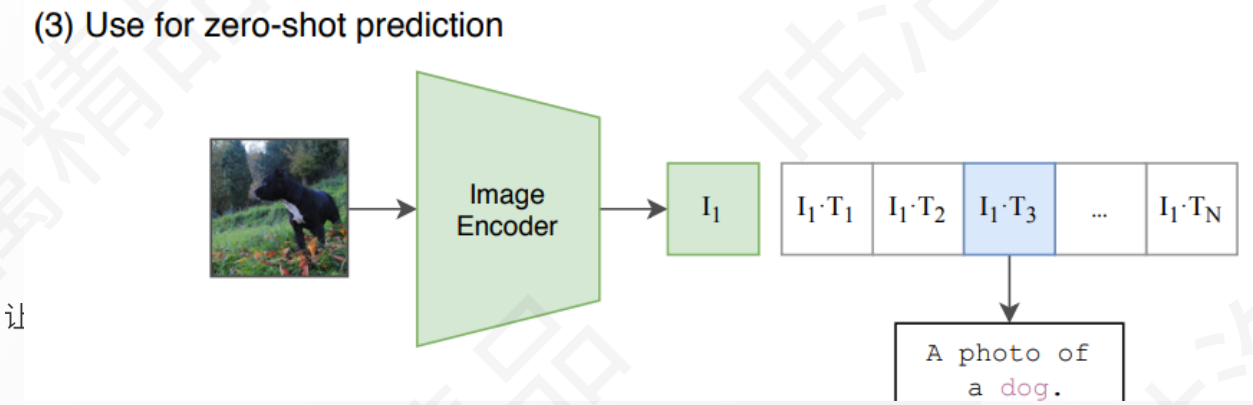
Contrastive Language–Image Pre-training(CLIP)

提取预测类别文本

由于CLIP 预训练时候的文本端输出输入的是个句子，但原始的类别都是句子，因此首先需要对文本类别进行一些单词转句子的处理，如法如下：使用 A photo of a {object}. 的提示模板 (prompt template) 进行构造，比如对于 dog，就构造成 A photo of a dog.，然后再送入 Text Encoder 进行特征提取，这样就会得到一个文本的特征向量。



推理预测



模型推理比较简单，只需要将输入图片传给ImagesEncoder模块，就会生成一个一维的图片特征向量，然后拿这个图片特征和 第二阶段生成的文本特征做余弦相似度对比，最相似的即为我们想要的那个结果，比如这里应该会得到 A photo of a dog.

咕泡科技 GUPAO TECH

Vision-Language Model

模型的训练基本包含两个阶段

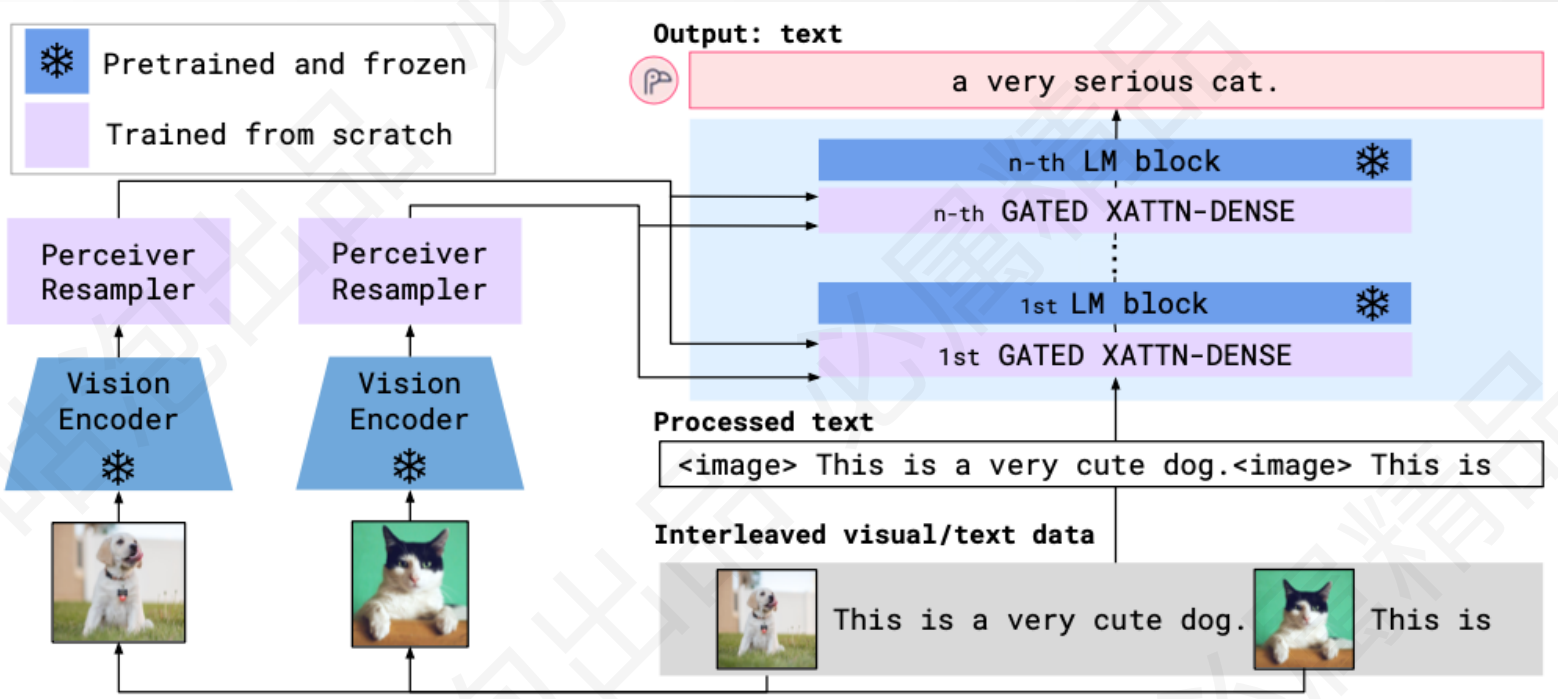
•预训练阶段（PreTraining）：通常是为了实现视觉特征与文本特征的对齐。有些模型的这一阶段也会分为两个子阶段，比如针对弱标签数据训练和人工标注训练，或者在训练中增加图片分辨率等

•微调阶段（Finetune）：此阶段通常是使用指令或特定任务数据进行微调，以增强模型遵循指令的能力和对话能力等，有些也会分位两个子阶段

Model	Date	PreTraining		Finetune		Remark
		Stage 1	Stage 2	Stage 1	Stage 2	
Flamingo-80B	22.04	ALIGN 1.8B + LTIP 312M 图文对 1536 TPUv4 * 15 天		M3W 185M 图像 + 文本 27M 短视频 + 文本		
BLIP-2 ViT-g FlanT5XL	23.01	129M 图文对 16 x A100-40G * 6 天	129M 图文对 16 x A100-40G * 3 天	N/A		
LLaVA-v1	23.04	595K 图文对		158K 图文对		
MiniGPT-v1	23.04	5M 图文对 4xA100-80G * 10 小时		3500 图文对 1xA100 * 7 分钟		
mPLUG-Owl	23.04	104B 图文 Token 32 x A100 * 7 天		392K 图文对 8 x A100 * 8 小时		
VisualGLM-6B	23.05	30M 中文图文对 + 300M 英文图文对		未明确		
Qwen-VL	23.08	1.5B 图文对(EN: 77, ZH: 23)	69M 图文对 + 7.8M文本	350K 图文对		
InternLM-XComposer-VL	23.09	1.1B 图文 + 67.7B 文本 Token(EN: 50.6 ZH: 17.1) + 10B 纯文本 Token		2.6M 图文对	260K 图文对	橙色表示根据 batch size 和 step 预估
LLaVA-v1.5	23.10	LCS 558K 图文对 8 x A100 * 6 小时		665K 图文对 8 x A100 * 20 小时		
MiniGPT-v2	23.10	38M 图文对 8 x A100 * 90 小时	3.2M 图文对 4 x A100 * 20 小时	81K 图文对 4 x A100 * 7 小时		橙色表示根据 batch size 和 step 预估

Flamingo

Flamingo: a Visual Language Model for Few-Shot Learning



1.Vision Encoder：作用是将图片或者视频帧转换为特征，作者采用 NFNet 的 F6 模型，输入图片，输出对应的特征，维度为 [S, d]，其中 S 表示有多少块，d 表示每一个块的特征维度。类似于其他 VLM 常用的 ViT 模型。

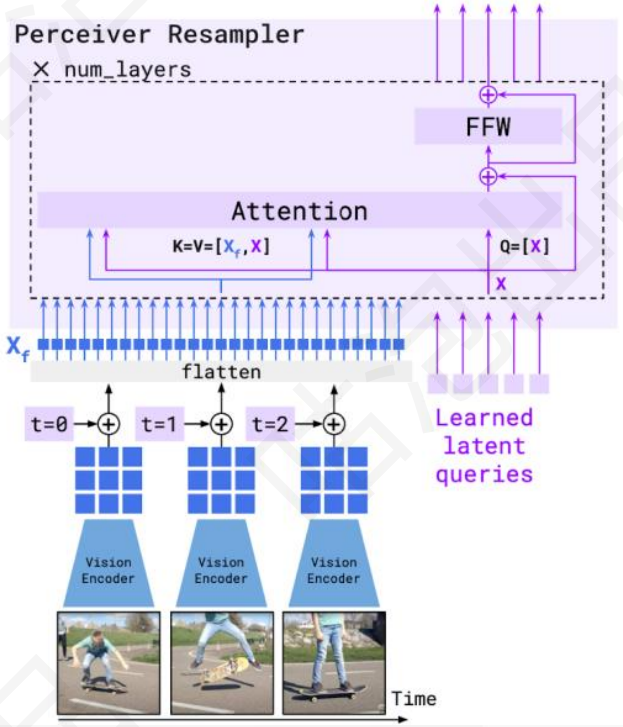
2.Perceiver Resampler：作用是对 Vision Encoder 生成的较大的图片特征转换为较小的 Visual Tokens，也就是进行采样，最后生成固定个数的 Token（64）。

3.Large Language Model：主要作用是接收 Viisual Token 和输入文本，然后生成文本

Flamingo

Flamingo: a Visual Language Model for Few-Shot Learning

- 每个图像经 Vision Encoder 会生成一个 [S, d] 的视觉特征，T 个图像对应 x_f 的维度为 [T, S, d]
- x_f 加上维度为 [T, 1, d] 的 time_embeddings
- 将时间和空间维度拉平，x_f -> [T*S, d]
- 将 x_f 作为 transformer block 的 Key 和 Value 输入
- 自定义的 R 个可学习的 Query Token，对应维度为 [R, d]
- 然后经过 num_layers 层 transformer block 得到对应的新的视觉特征 x，维度为 [R, d]，和可学习的 Query 维度一致



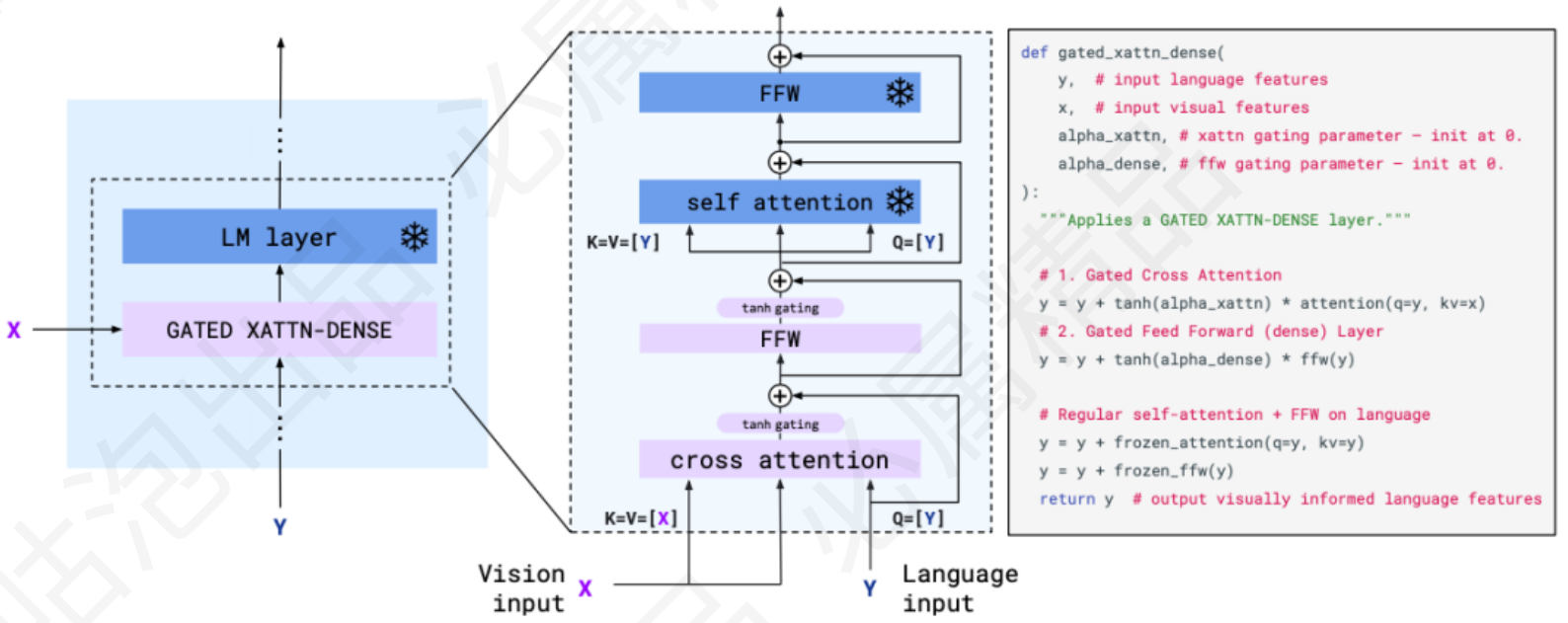

```
def perceiver_resampler(
    x_f,  # The [T, S, d] visual features (T=time, S=space)
    time_embeddings,  # The [T, 1, d] time pos embeddings.
    x,  # R learned latents of shape [R, d]
    num_layers,  # Number of layers
):
  """The Perceiver Resampler model."""

  # Add the time position embeddings and flatten.
  x_f = x_f + time_embeddings
  x_f = flatten(x_f)  # [T, S, d] -> [T * S, d]
  # Apply the Perceiver Resampler layers.
  for i in range(num_layers):
    # Attention.
    x = x + attention_i(q=x, kv=concat([x_f, x]))
    # Feed forward.
    x = x + ffw_i(x)
  return x
```

Flamingo

Flamingo: a Visual Language Model for Few-Shot Learning

通过 Cross Attention 实现视觉特征和文本特征的交叉

- 将 **Perceiver Resampler** 生成的 Vision input 作为 Key、Value 输入，Language input 作为 Query 输入
- 首先经过 Gated Cross Attention
- 然后经过 Gated FFW（Feed Forward MLP）
- 输出并作为下一个 LLM 的 transformer layer 的输入




```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter - init at 0.
    alpha_dense, # ffw gating parameter - init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 1. Gated Cross Attention
  y = y + tanh(alpha_xattn) * attention(q=y, kv=x)
  # 2. Gated Feed Forward (dense) Layer
  y = y + tanh(alpha_dense) * ffw(y)

  # Regular self-attention + FFW on language
  y = y + frozen_attention(q=y, kv=y)
  y = y + frozen_ffw(y)
  return y  # output visually informed language features
```

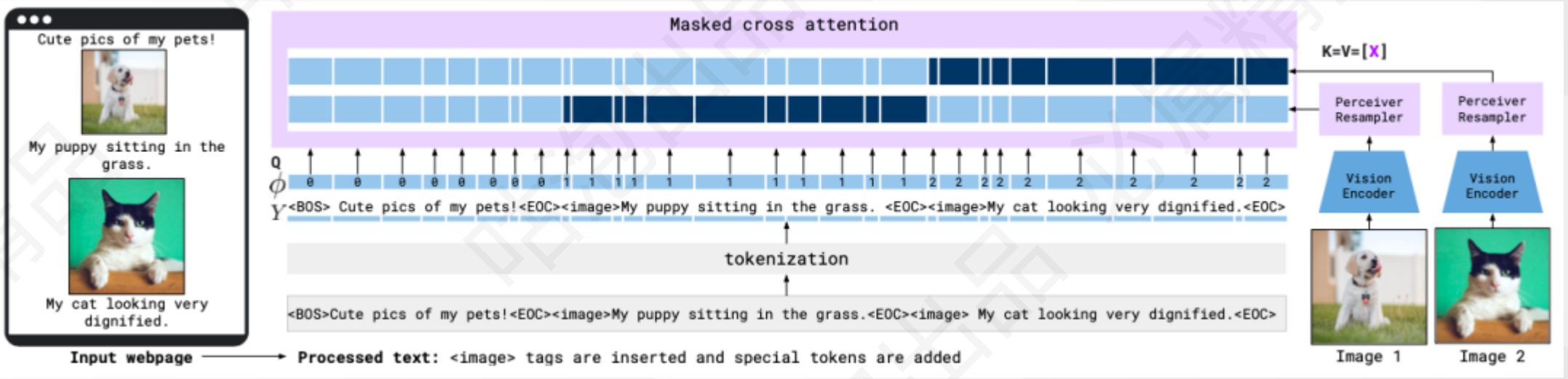
咕泡科技 GUPAO TECH

Flamingo

Flamingo: a Visual Language Model for Few-Shot Learning

多个图像、文本输入的排布

•视觉图像全部需要经过 **Vision Encoder** + **Perceiver Resampler** 生成的 Vision input 作为 Key、Value 输入。
•文本全部经 Tokenization 后输入。当然，在文本中会插入 <BOS>、<EOC> 等起止 Token，也会插入 <image> Token 作为图像的位置标识。
•其中的 Cross Attention Mask 也经过特殊设计，让文本只和相关图像进行交互。



Flamingo

Flamingo: a Visual Language Model for Few-Shot Learning

数据及处理

•M3W（MultiModal Massive Web）：交替的图像文本数据集。从 43M 个网页中提取的文本和图像。对于每一个文档，随机选择一个 L=256 Tokens 的序列，并且选择文档中的前 N=5 个图像，总共 185M 图像。
•图像-文本对：
- 使用 ALIGN 数据集（[2102.05918] Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision），包含 1.8B 图像-文本对
- 312M 图像-长文本对的 LTIP数据集，目标是更好的质量和更长的描述。

•视频-文本对：一个视频-文本对数据集，包含 27M 个短视频（平均 22 秒），并配有句子描述

数据清洗和去重

•删除非英文文档、低质文档以及重复文档
•删除太小的图像，比如长或宽小于 64 像素
•删除太宽或太窄的图像，比如长宽比例大于 3 的图像
•删除低质图像，比如只有一种颜色
•删除经过清洗后不包含图像的文档

Flamingo

Flamingo: a Visual Language Model for Few-Shot Learning

预训练和微调

Vision Encoder 预训练

Vision Encoder 基于 ALIGN 数据集 和 LTIP 数据集在 512 个 TPUv4 上预训练，训练时图像分辨率为 288x288，embedding 空间大小为 1376，batch size 为 16,384。经过 1.2M 个训练 step，每个 step 包含两次梯度计算

Flamingo 预训练

总共包含 3 个模型，Flamingo-3B、Flamingo-9B 和 Flamingo-80B。所有模型都是使用 JAX 和 Haiku 实现，所有训练和评估均在 TPUv4 上运行。此阶段使用 M3W 数据集和视频-文本对数据集。其中，**Flamingo-80B 在 1536 个 TPUv4 上训练了 15 天**

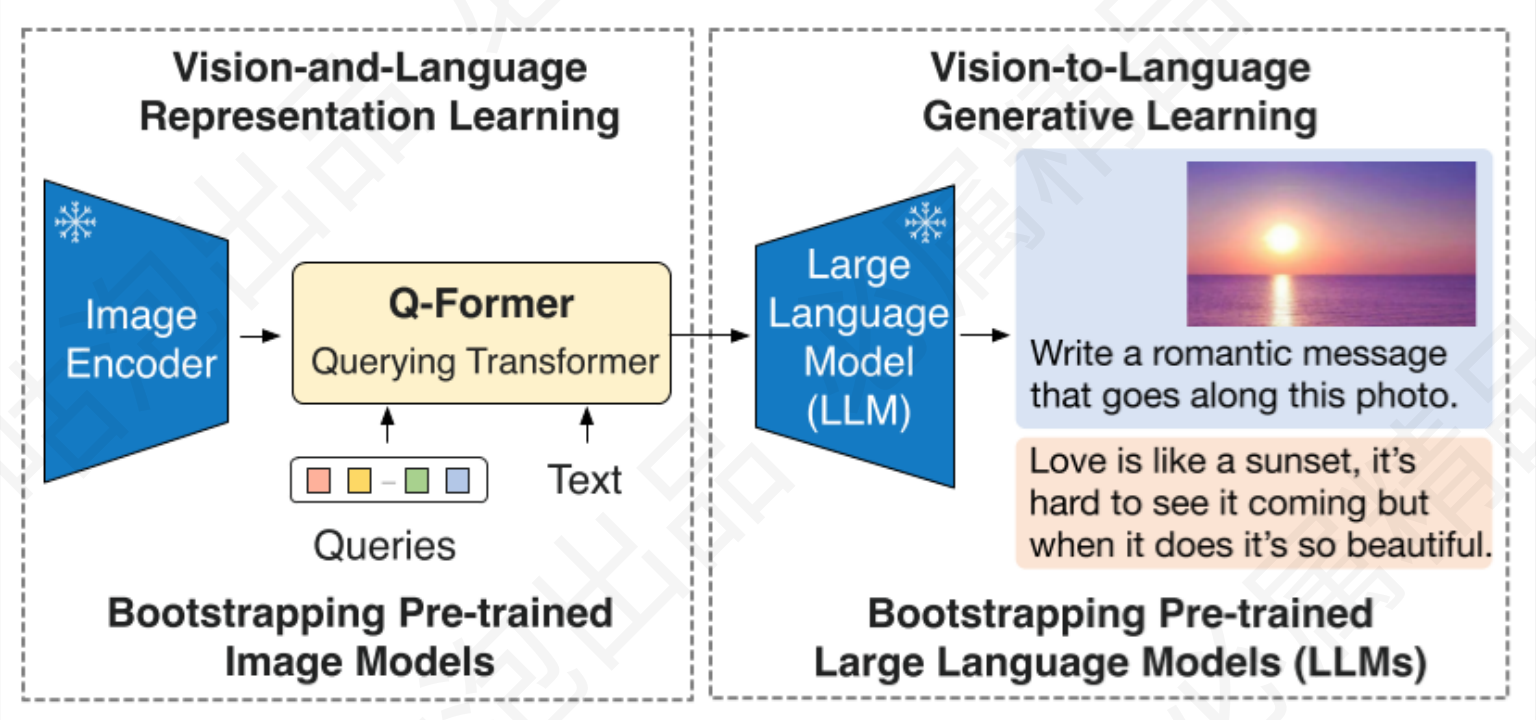
- 所有 Embedding、Self-Attention、Cross-Attention、FFW 层都在切分为 16 个 shard，执行模型并行，NFNet Vision Encoder 未切分
- 采用 Zero1 对优化器状态进行分片
- 训练中模型参数、优化器累加都是用 FP32，激活和梯度使用 BF16
- 输入图像分辨率从预训练的 288x288 扩大到 320x320

微调

在微调 Flamingo 时，作者会冻结基础的 LM 层，并训练与预训练期间相同的 Flamingo 层，同时将输入图像分辨率从 320x320 扩大到 480x480。与预训练不同的是，此阶段也会对 Vision Encoder 进行微调，这通常能改善效果。

BLIP-2

BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models



1.Image Encoder：和 Flamingo 模型的 Vision Encoder 作用一样，用于提取视觉特征，采用的是CLIP ViT-L/14 和 EVA-CLIP ViT-g/14

2.Q-Former：Query Transformer，用来弥补 image 模态和 text 模态的差距，实现特征对齐

3.Large Language Model：和 Flamingo 模型的 LLM 作用相同，用于生成文本，没有对 LLM 的结构进行修改

咕泡科技 GUPAO TECH

BLIP-2

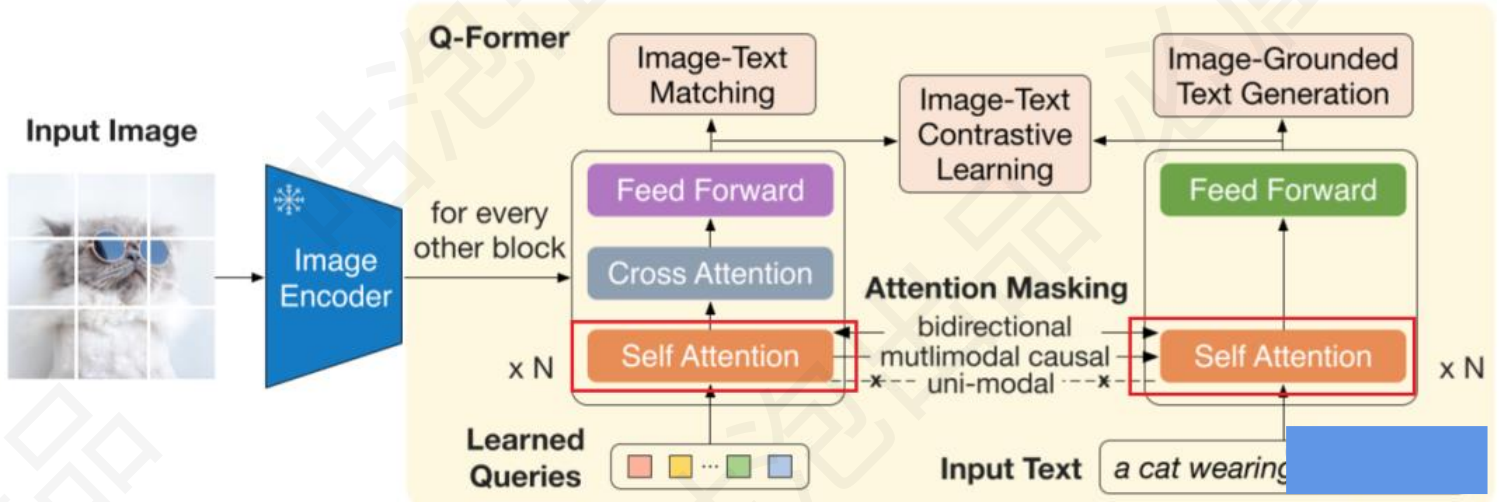
BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models

Q-former

从 Image Encoder 中提取**固定数量的输出特征**，**与输入图像分辨率无关**。其由两个共享 Self Attention 的 Transformer 子模块组成

- Q-Former 左侧为 image transformer：与冻结的 image encoder 交互以进行视觉特征提取
- Q-Former 右侧为 text transformer：可以用文本 encoder 和 文本 decoder

可学习的 Query embedding 作为 image transformer 的输入。这些 Query embedding 在 Self Attention 层相互交叉，并通过 Cross attention 层（每隔一个 transformer block 有一个 Cross attention）与冻结的 image encoder 输出的 image embedding 进行交叉。此外，这些 Query embedding 还通过相同的 Self Attention 与 text embedding 相交叉。使用 Bert Base 的预训练权重来初始化 Q-Former，其中的 Cross Attention 是随机初始化的，Q-Former 总共包含 188M 个参数（包括 Query embedding）

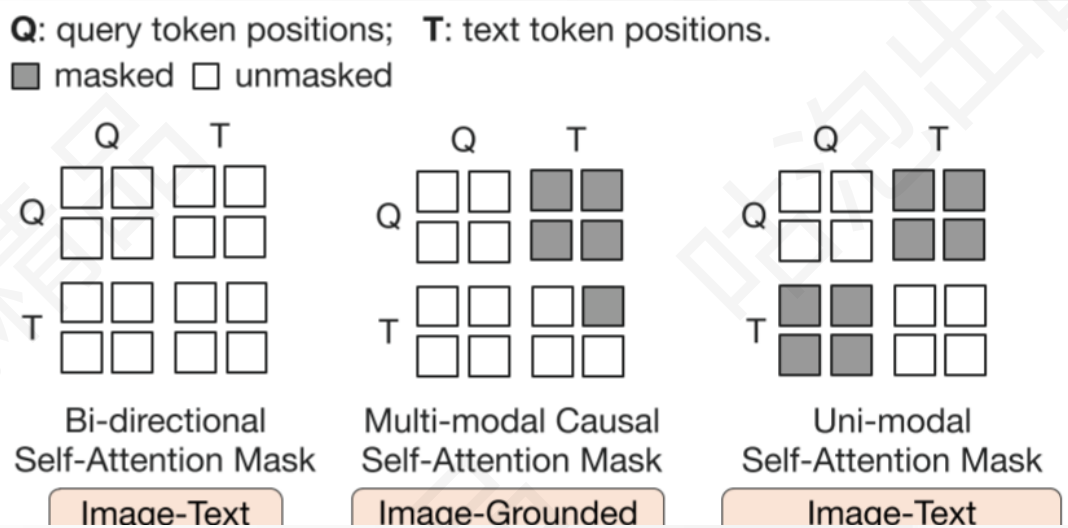


BLIP-2

BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models

使用不同的 Self Attention Mask 来控制 Query embedding 和 text embedding 的交互

- Image-Text Matching：Query 中的每个 Token 和 Text 中的每个 Token 都能看到 Query + Text 中的所有 Token。此时的 **text transformer 相当于 encoder**
- Image-Grounded Text Generation：Query 会 Mask 掉所有 Text，Text 有 Causal Mask，Query 中的 Token 能看到 Query 内的所有 Token，而看不到 Text 中的 Token；同时，Text 中的 Token 都能看到所有 Query 中的 Token，并且只能看到 Text 中当前 Token 之前的 Token。此时的 **text transformer 相当于 decoder**
- Image-Text Contrastive Learning： Query 和 Text 都 Mask 掉彼此，而在内部没有 Mask，Query 中的 Token 只能看到 Query 中的所有 Token，Text 中的 Token 只能看到 Text 中的所有 Token，此时 **text transformer 相当于 encoder**



使用了 32 个 Query，每个 Query 的维度为 768，与 Q-Former 中的 hidden 维度相同。也就是对应的 Query 的维度为 （32 x 768），由于 transformer block 并不会更改输入的维度，因此 image transformer 输出的维度 Z 也为（32 x 768），这相比冻结的 image encoder 输出的维度小得多（比如，ViT-L/14 对应输出维度为 257 x 1024）。这种架构与预训练一起协同，迫使这些 Query 提取与 Text 最相关的视觉信息

让每

BLIP-2

BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models

模型预训练和微调

Vision Encoder 预训练

对于冻结的 image encoder，选择 CLIP 的 ViT-L/14 以及来自 EVA-CLIP 的 ViT-g/14
对于其中的 ViT 模型，删除了其最后的一个 transformer block，可以稍微提升性能

第一阶段表征学习阶段

将 Q-Former 连接到冻结的 image encoder 上（没有 LLM），并使用图像-文本对进行预训练：
•预训练 250K step
•图像分辨率 224x224
•ViT-L 和 ViT-g 对应的 batch size 分别为 2320 和 1680
•ViT-L 和 ViT-g 都使用 FP16
•单台 16 x A100-40G 机器，更大的 ViT-g 需要不到 6 天

第二阶段预训练阶段

将 Q-Former（带有冻结的 image encoder）连接到冻结的 LLM，以获得 LLM 强大的语言生成能力：
•冻结的 LLM：
- Encoder-Only 模型，作者采用了无监督训练的 OPT 系列模型
- Encoder + Decoder 模型，作者采用了经过指令训练的 FlanT5 模型

•图像分辨率 224x224
•预训练 80K step
•OPT 和 FlanT5 对应的 batch size 分别为 1920 和 1520
•OPT 使用 FP16，FlanT5 使用 BF16
•单台 16 x A100-40G 机器，最大的 FlanT5-XXL 需要不到 3 天

咕泡科技
GUPAO TECH

LLaVA-v1

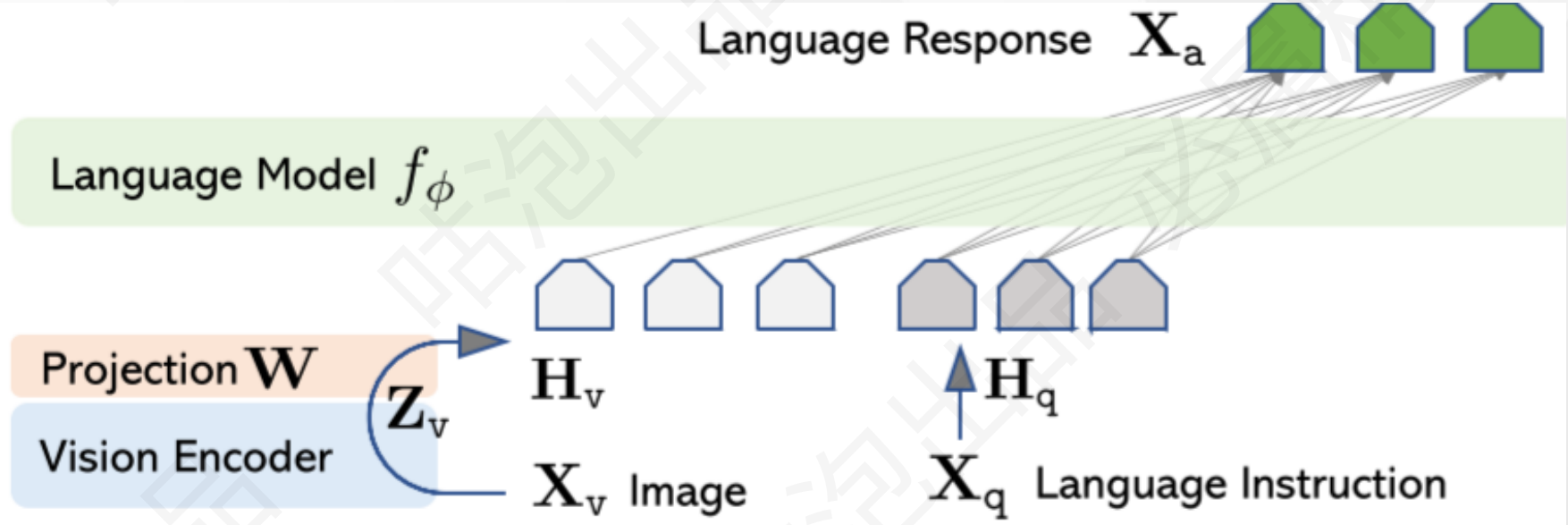
Visual Instruction Tuning

模型结构

1.Vision Encoder：和 Flamingo 模型的 Vision Encoder 作用一样，也是用于提取视觉特征，作者采用的是CLIP ViT-L/14

2.Projection W：其比 Flamingo 中的 **Perceiver Resampler**和 BLIP-2 中的 **Q-Former** 简单得多，只是一层简单的 Linear，将 image feature 映射到 LLM 的 word embedding 空间

3.Large Language Model：和 Flamingo 模型的 LLM 作用相同，用于生成文本，不过作者没有对 LLM 的结构进行修改，直接使用了 Vicuna-v1.5 13B 模型



LLaVA-v1

Visual Instruction Tuning

预训练和微调

特征对齐预训练

使用 Spacy 在整个 CC3M 数据集上为每个描述提取名词短语，并计算每个唯一名词短语的频率。跳过频率小于 3 的名词短语，因为它们往往是比较罕见的组合概念和属性，已被其他描述覆盖。从剩余频率最低的名词短语开始，将包含此名词短语的描述添加到候选库中。如果名词短语的频率大于 100，会从中选择一个大小为 100 的子集。经过这个过程，最终筛选出 595K 图像-文本对用于预训练。Vision Encoder 和 LLM 都冻结，只训练 Projection 层。

端到端微调

在 MultiModal Chatbot 和 Science QA 两个场景进行了微调，微调期间会保持 Visual Encoder 冻结，Projection 层和预训练的 LLM 都会更新。其中 MultiModal Chatbot 是作者收集的数据集，包含 158K 的语言-图像指令微调样本：

58K 对话

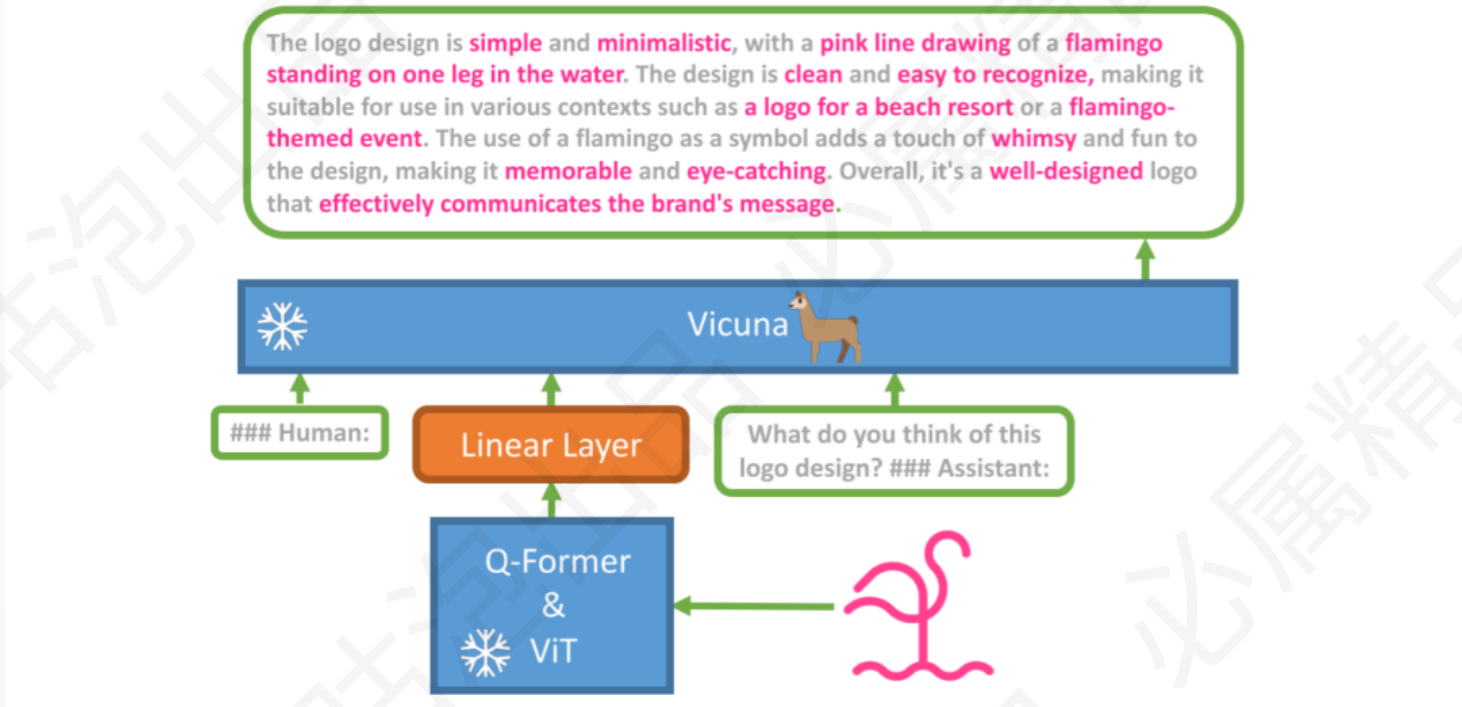
23K 详细描述

77K 复杂推理

MiniGPT-v1

MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models

模型结构



1.Vision Encoder：直接使用了 BLIP-2 的方案，作者用的是EVA-CLIP ViT-G/14
2.Projection：BLIP-2 的 Q-Former 也完整保留，同样后面增加了一层可训练的 Linear 层
3.Large Language Model：使用 Vicuna-v0 模型作为 LLM

MiniGPT-v1

MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models

模型预训练和微调

特征对齐预训练

在整个预训练阶段，Vision Encoder + Q-Former 和 LLM 都保持冻结状态，只训练线性投影层。作者使用了 Conceptual Caption 数据集，SBU 数据集和 LAION 数据集的集合来训练，batch-size 为 256，经过 20,000 个 step，大概覆盖了 5M 个图像-文本对。整个过程在 4xA100 80G GPU 上经过 10 个小时训练完成

端到端微调

在微调阶段，从Conceptual Caption 数据集中随机选择了 5,000 个图像，然后用预训练的模型为每个图像生成了一个文本描述，之后在经过一系列处理，最终挑选出 3,500 左右的图像-文本对构成微调阶段的数据集。基于此，只需训练 400 个 step，batch size 为 12，在单个 A100 上 7 分钟即可完成训练

咕泡科技
GUPAO TECH

下次课预告

多模态大模型实战-VQA

To do list:

1. 阅读BLIP,ViT,CLIP原文