基于多种多模态融合方式的图生文与文生图模型

龚舒凯　徐少轩　赵嘉浩

日期：2025 年 7 月 22 日

完整实验代码见https://github.com/xushaoxuan123/Multimodal-Machine-Learning

1　图生文

1.1　基于交叉注意力 (Cross Attention) 机制的图像-文本对齐模块

[12] 提出了一种基于交叉注意力机制方法对齐图像和文本模态的方法。如图1所示，在得到了图像的表示 $\mathbf{P} \in \mathbb{R}^{(H\times W)\times C}$ 和文本的表示 $\mathbf{T} \in \mathbb{R}^{n\times D}$ 之后，我们首先让图像通过一个线性层 $\mathbf{W}_{\text{img}} \in \mathbb{R}^{C\times d}$ 映射为键 $\mathbf{K}_{\text{img}} = \mathbf{W}_{\text{img}}\mathbf{P} \in \mathbb{R}^{(H\times W)\times d}$ 和值 $\mathbf{V}_{\text{img}} = \mathbf{W}_{\text{img}}\mathbf{P} \in \mathbb{R}^{(H\times W)\times d}$，并让文本通过另一个线性层 $\mathbf{W}_{\text{text}} \in \mathbb{R}^{D\times d}$ 映射为查询 $\mathbf{Q}_{\text{text}} = \mathbf{W}_{\text{text}}\mathbf{T} \in \mathbb{R}^{n\times d}$。然后按照如下方式计算文本和图像之间的交叉注意力：

$$\text{CrossAttn} = \text{softmax}(\mathbf{Q}_{\text{text}}\mathbf{K}_{\text{image}}^{\top})\mathbf{V}_{\text{image}} \in \mathbb{R}^{n\times d} \tag{1}$$

这种机制可以在文本和图像特征之间建立一对一的比较，确定每个文本特征与特定位置的图像特征之间的关系，从而将图像信息融入到文本信息中，服务于后续的图生文。

文本的表示 $\mathbf{T} \in \mathbb{R}^{n\times D}$ 可以通过预训练的 BERT[1] 或 CLIP[7] 的文本编码器获取。图像的表示 $\mathbf{P} \in \mathbb{R}^{(H\times W)\times C}$ 有两种获取方式：(1) 使用 ResNet[4] 这样的 CNN 网络作为特征提取器，将展平后的特征图的每个像素视作 token，形成序列状的图像的表示；(2) 使用 CLIP 这样的 ViT[2] 网络作为特征提取器，模型的输出直接是序列状的图像表示。

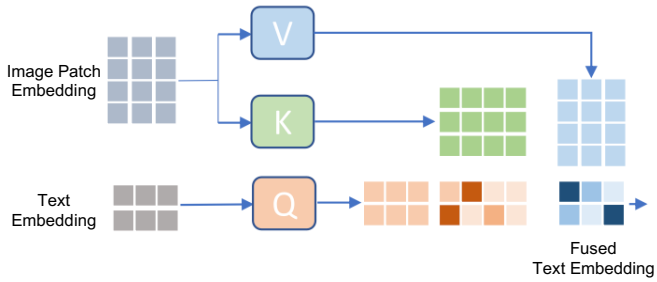


图 1: 交叉注意力机制原理

1.2　基于 Q-former 的图像-文本对齐模块

[5] 提出了一种轻量级的 Transformer 模型 Q-former（Querying Transformer），使用可学习的查询向量从冻结的图像编码器中提取对文本生成最有用的视觉特征。

如图2所示，在模型架构上，queries 相当于 Q-former 的学习参数，目的是为了更好的提取视觉信息中与文本相似的特征。此外，自注意力层和前馈层的共享参数也确保了模型提取特征与文本特征的相似程度。图片信息通过冻结的视觉编码器，通过 cross-attention 机制与 queries 进行交互。

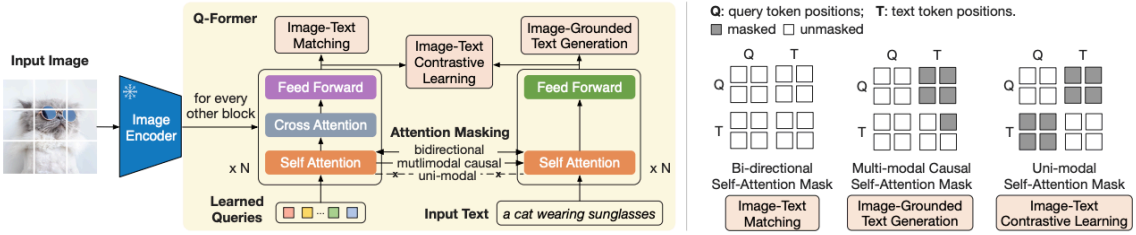


图 2: Q-former 模块原理

如图8所示，论文将 Q-former 作为多模态大模型 adapter 的训练分为了两个阶段：

1. **使用图片-文本对在三个任务上训练 Q-former**。为了让 Q-former 学习到在视觉-文本任务上更加泛用的特征，作者使用了 ITM，ITC 和 ITG 三种任务对其进行预训练。其中，ITM 任务是判断文本与图片是否匹配，采用双向的注意力机制。ITG 任务是文本生成任务，其中的 Query 作为已知信息不使用 mask，只 mask 文本信息进行自回归生成。ITC 任务是对比学习任务，要求两个模态不能关注相互的信息，因此是单模态自注意力机制。
2. **与大模型进行对齐**。在 Qformer 预训练阶段完成后，需要将与文本特征相似的特征空间与大模型自身的文本空间进行对齐。在这一部分的训练中，视觉编码器和大模型本身是冻结的，仅对 Q-former 和全连接层进行微调，将生成的特征以类似 prompt 的形式传入大模型并生成文本。

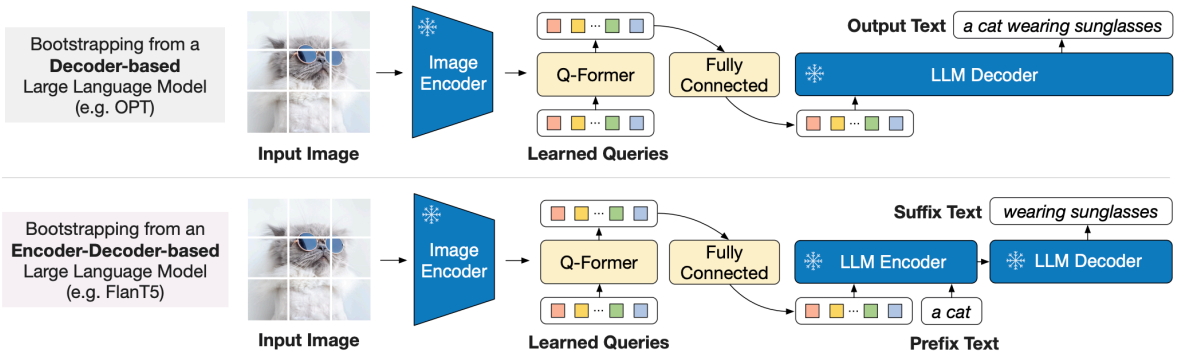


图 3: Q-former 模块与大模型的对齐

值得注意的是，Qformer 的预训练任务中已经包含了图生文任务，虽然 Q-former 自身是一个类 BERT 的架构，但是也可以通过将初始 token 从 <CLS> 切换为 <DEC> 进行自回归生成。因此，我们也尝试使用 Qformer 架构直接作为文本生成模型进行训练。

1.3 实验

1.3.1 数据集

我们只使用提供的 Flickr8k 数据集进行实验。由于 Flickr8k 数据集中一张图像对应 5 条文本，为避免模型混淆同一图像的多种表述，我们采用了单一标注策略，即每张图片仅对应一条文本，以减少模型对于同义不同表述文本的歧义识别。经过处理后，我们获得了 8091 个图像-文本对，并按照 8:2 的比例将其划分为训练集和验证集。

1.3.2 实验设置

如表1所示，基于交叉注意力机制和 Q-Former 模块，我们设计了四种模型结构。

视觉解码器：在第一组实验中，我们只保留 ResNet 的卷积层部分，将其输出的特征图 ($7 \times 7 \times 512$) 展平为 49×512 的向量 [11]，通过一个全连接层将其从图像向量空间映射到文本向量空间，获得序列状的 image tokens；而在后三组实验中，我们直接将 CLIP 的输出通过一个全连接层映射到文本向量空间。在所有的实验中，视觉编码器都是冻结的。

模态对齐模块：在前两组实验中，我们按照前文所述使用交叉注意力模块时，以 image tokens 作为 query，text tokens 作为 key 和 values 实现模态的交互。在后两组实验中，我们既用提供的 Flickr8k 数据集完整训练了一个没有预训练权重的 Q-former，也在 [5] 提供的预训练权重上微调了一个 Q-former。

文本编码器：前两组实验中，我们使用 6 层的 Transformer Decoder 作为文本生成器。在第三组实验中，我们将 Q-former 中的 BERT 结构进行修改，将其初始 token 从 <CLS> 切换为 <DEC>，使之能够进行自回归生成。在第四组实验中，我们希望发挥预训练大模型在文本生成上的强大能力，使用冻结住的 TinyLlama-1.1B-Chat-v1.0[13] 作为文本编码器。TinyLlama 是一个仅有 1.1B 参数的语言模型，但其不具备多模态能力。因此，我们通过微调 Q-former，使 Q-former 学习出来的 queries 既包含图像信息，又能与 TinyLlama 的文本空间对齐。最后，我们在训练和推理过程中还加上了一个简短的文本 prompt 指导模型的生成 (如：*"Generate a simple short caption:"*)，与学习得到的 queries 拼接后输入 TinyLlama，指导其进行文本的生成。

视觉编码器	模态对齐模块	文本解码器	隐藏层大小	总参数量	可训练参数量
ResNet-18	Cross Attention	Transformer Decoder	512	34.2M	21.5M
CLIP(ViT-B/32)	Cross Attention	Transformer Decoder	768	108M	21.5M
CLIP(ViT-g/14)	Q-former	BERT	768	1121M	110M
CLIP(ViT-g/14)	Q-former (预训练)	TinyLlama-1.1B	2048	2274M	188.3M

表 1: 图生文模型架构设置

1.3.3 实验结果

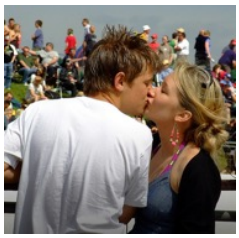
我们使用 BLEU(Bilingual Evaluation Understudy) 评估图生文的质量。BLEU 通过比较生成文本和一组参考文本之间的 n 元组 ($n = 1, 2, 3, 4$) 的重合程度来衡量生成质量。在评估模型的性能

时，我们将每张测试图像所生成的文本与其实际对应的五条参考文本进行对比，以量化生成文本的质量。实验结果如表2所示。

视觉编码器	模态对齐模块	文本解码器	BLEU1	BLEU2	BLEU3	BLEU4
ResNet-18	Cross Attention	Transformer Decoder	22.7	13.9	8.7	5.2
CLIP(ViT-B/32)	Cross Attention	Transformer Decoder	23.3	15.5	10.2	6.2
CLIP(ViT-g/14)	Q-former	BERT	52.3	36.7	24.7	15.8
CLIP(ViT-g/14)	Q-former (预训练)	TinyLlama-1.1B	**72.2**	**50.9**	**36.1**	**25.9**

表 2: 文生图模型在 Flickr8k 数据集上的表现对比

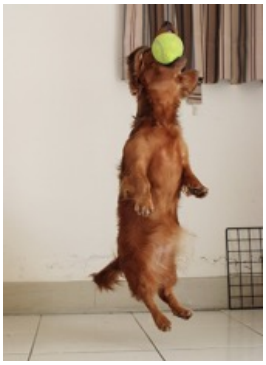
在所有的四组实验中，结合了微调的 Q-former 和 TinyLlama 的文生图模型获得了最佳的表现，在 BLEU1 到 BLEU4 四个指标上均取得最高值。从实验数据中可以发现 (1) 相比使用简单的交叉注意力机制模块，使用 Q-former 模块的图生文模型在生成质量上有着显著的提高。这是因为 Q-former 不仅使用交叉注意力机制，还使用对比学习和生成任务进一步加强了文本模态和图像模态的对齐。(2) 相比使用预训练的 ResNet 编码器，使用 CLIP 编码器的模型在生成质量上有部分提升。这是因为 CLIP 在大规模预训练的过程中已经通过对比学习文本模态进行对齐，生成的图像表征质量更高。(3) 尽管我们冻结了 TinyLlama，但在经过图像-文本模态对齐后，其生成表现仍然显著优于可训练的 Transformer Decoder 和 BERT。



Generated: **A couple kissing in front of a crowd.**

Generated: **A man is holding a bird in his hand.**

Generated: **Two brown dogs are playing with each other in the grass.**



Generated: **A brown dog catches a tennis ball in the air.**

Generated: **A girl in a bikini is lying on the sand.**

Generated: **Two women in red dresses are standing next to each other.**

图 4: 微调 Q-former+TinyLlama 图生文模型效果展示

2 文生图

2.1 Lora 微调 Stable Diffusion

使用的预训练模型是 SDXL-base-1.0，微调使用的代码是 huggingface 的diffusers仓库下 examples 中的 text_to_image 中的 train_text_to_image_lora_sdxl.py，我们使用时具体的参数配置见 finetune_diffusion.sh，其中，微调使用的数据构造方式如下，将 Flickr 8k 的 caption.txt 中出现的前 7000 张图片，以及每个图片对应的第一个 caption 构造成 < 图片，对应描述 > 的 pair。

(下面提到的所有测试都没有在这 7000 张图上进行，选取的都是剩下的 1091 个图片和它对应的描述。)

尝试的过程中我们遇到了一些问题，比如最初 lora 的 rank 只开了 4，训练步数也不够多，生成的时候步数不够多，生成了大量混乱的图片，比如一个人有很多四肢等让人生理不适的场面。经过一些实验之后最终我们用了 128 的 rank，生成的时候采样 100 步。我们展示的图片均使用这种配置生成 (生成使用的代码见 generate_diffusion.py)。

为了观察我们微调的效果，我们对没有微调的 sdxl 和微调之后的 sdxl 生成图片的效果进行了一些对比，比如

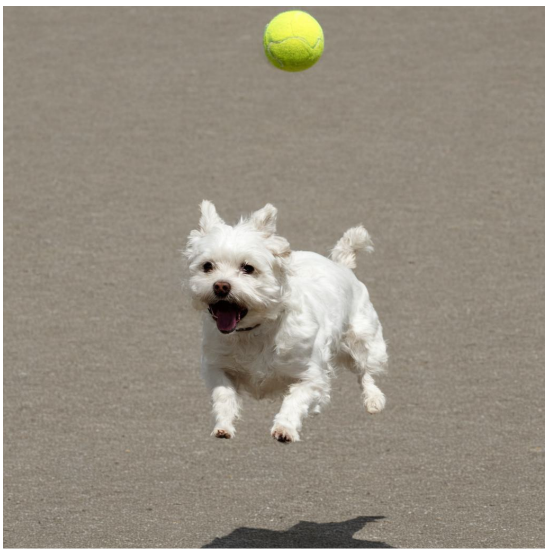
图 5: Stable Diffusion 微调前后结果对比

没有在 Flickr 8k 上微调之前，生成的是比较漫画风格的蝙蝠侠，而 Flickr 8k 中大多是真实世界中的图片，从结果来看，stable diffuison 生成的风格确实和 Flickr 8k 更接近了!

Stable Diffusion 微调后生成效果展示

这里仅展示 6 张，更多生成结果见 res_with_captions_sdxl_finetuned.zip, 同时我们也使用相同的文本，在去掉 Lora 块的情况下直接使用预训练好的 Stable Diffusion 生成了图片，具体结果见 res_with_captions_sdxl_ori.zip

A human figure stands on the peak of a snowy mountain .



A small white dog jumping to catch a tennis ball .

A person in the distance hikes among hoodoos with stars visible in the sky .

A man on the beach in jeans looking at his camera .

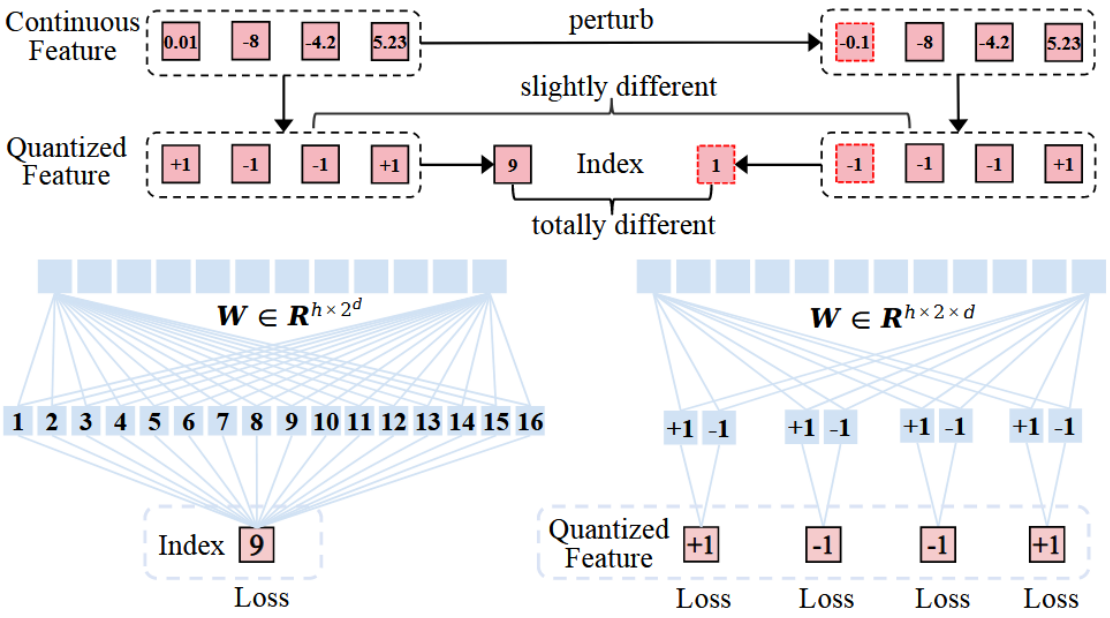
A girl in an orange tank top stands outside a cafe .

a blond boy petting a tiger

2.2 Infinity

Infinity[3] 由字节跳动在近期发布，是一个基于位级别视觉自回归建模的文本到图像生成模型，将传统的索引级别标记替换为位级别标记，显著提升了生成能力和细节表现。模型包含三个主要组件：

1. Bitwise Visual Tokenizer



相较于传统的 tokenizer 使用一个词表，然后通过找最相似的向量，来给这个特征向量一个编号，这里的 bitwise tokenizer 会把一个特征向量直接量化，比如二值球面量化 (Binary Spherical Quantization, BSQ)：

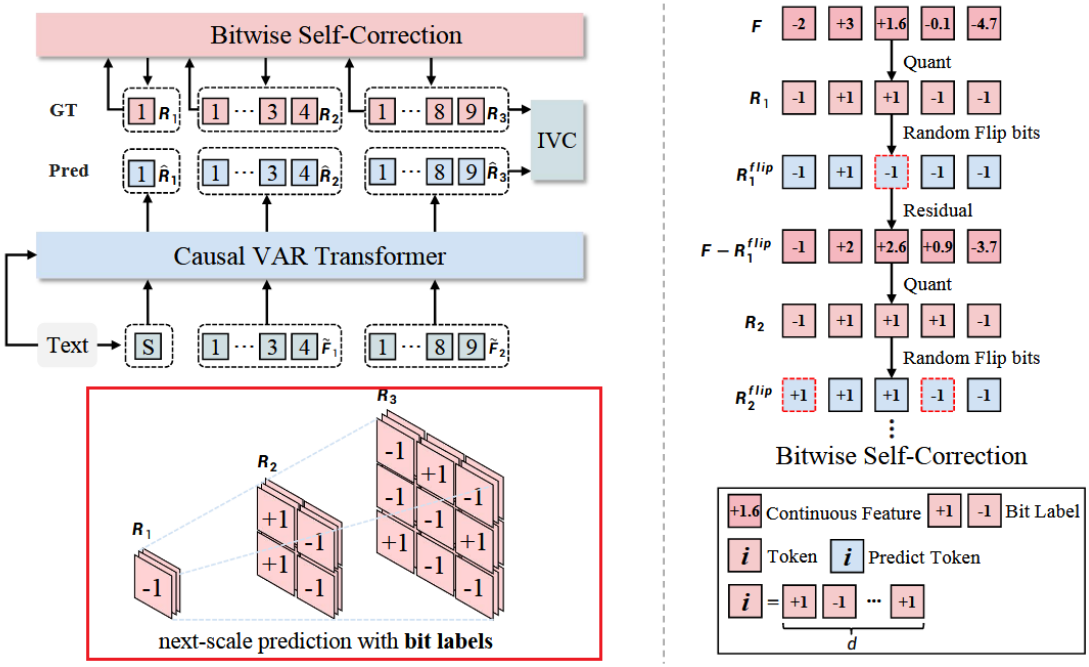
$$q_k = Q(z_k) = \frac{1}{\sqrt{d}} \text{sign}\left(\frac{z_k}{|z_k|}\right) \tag{2}$$

其中 z_k 是输入特征，d 是特征维度。

在进行分类的时候如上图所示，也不用传统的分类器，而是按 bit 预测的 IVC 分类器。

通过实际使用以及对论文的阅读，我们理解：这个设计是 infinity 比较独道的地方，大大降低了词表增大时硬件的负担，相比于传统的词表大小，这里的词表几乎是无限大了，infinity 这个模型的名字也正是来源于此。

2. Multi-scale Generation



生成的整体架构继承自 Var，把文本 encoder 输出的 embedding 当作初始位置，然后开始逐步预测，每次预测的图片尺寸都会变大，直观上感受是一个从整体到细节的过程。

$$p(R_1, ..., R_K) = \prod_{k=1}^{K} p(R_k | R_1, ..., R_{k-1}, \Psi(t)) \tag{3}$$

其中：

R_k 是第 k 个尺度的残差 (随着 k 的怎加尺寸在不断变大)

$\Psi(t)$ 是文本特征

特征重建通过累积残差实现：

$$F_k = \sum_{i=1}^{k} \text{up}(R_i, (h, w)) \tag{4}$$

3. Text Conditioning

Infinity 使用 T5 来生成文本 embedding，并通过多种方式影响生成：

- 交叉注意力机制
- 条件自适应层归一化
- Classifier-free Guidance

4. 结果展示

我们使用的是 2B 规模的 Infinity，这里展示六张，更多生成结果见 res_with_captions_inf.zip，这里展示 6 张。

A human figure stands on the peak of a snowy mountain .

A small white dog jumping to catch a tennis ball .

A person in the distance hikes among hoodoos with stars visible in the sky .

A man on the beach in jeans looking at his camera .

A girl in an orange tank top stands outside a cafe .

a blond boy petting a tiger

5 一些感受

1. 相较于 diffusion 过程要逐步的 sample，infinity 使用的范式确实更快，比如我们实验的过程中用 stable diffuison 生成 1000 张图片大约需要 3h(nvidia A6000 单卡)，infinity 只需要 50min 左右，确实如 infinity 的论文中所说相较 stable diffusion 快了 3 倍。

2. 由于我们没有微调，所以生成的风格并不是很“Flickr 8k”，比如上面展示的那只老虎看起来就非常的“ai”。

3. 人的手指，一些动物的某些部位（比如狗的嘴）。还是比较频繁出现问题。

2.3 Our Diffusion

这里训练用的数据和上述 Lora 微调 Stable Diffusion 时的相同，我们用 google 预训练的bert-base作为文本的 encoder，U-Net[10] 做为 backbone，训练及模型细节的代码见 train_my_diffusion.py。下面讲述引入文本信息的方法：

首先，我们使用预训练的 BERT 模型对输入的文本信息进行 encoding

为了匹配 UNet 中交叉注意力层的维度，我们对 BERT 的输出 embedding 过一个线性层（由于显存有限，我们在训练过程中没有训练 bert 主体的参数，而是只训练这个线性层参数)：

Listing 1: 文本嵌入投影

```
self.text_projection = nn.Sequential(
    nn.Linear(text_encoder_dim, cross_attention_dim),
    nn.GELU(),
    nn.LayerNorm(cross_attention_dim)
)
text_embeds = self.text_projection(text_embeds)
```

相较于经典的 Unet 结构，UNet 的部分模块被我们替换为交叉注意力层，以便能够融合文本嵌入信息。

Listing 2: UNet 配置

```
self.unet = UNet2DConditionModel(
    sample_size=64,
    in_channels=3,
    out_channels=3,
    layers_per_block=2,
    block_out_channels=(64, 128, 256, 256),
    down_block_types=(
        "CrossAttnDownBlock2D",
        "CrossAttnDownBlock2D",
        "DownBlock2D",
        "DownBlock2D",
    ),
    up_block_types=(
        "UpBlock2D",
        "UpBlock2D",
```

```
        "CrossAttnUpBlock2D",
        "CrossAttnUpBlock2D",
    ),
    cross_attention_dim=cross_attention_dim,
)
```

其中，`CrossAttnDownBlock2D` 和 `CrossAttnUpBlock2D` 模块负责在下采样和上采样过程中引入文本信息，通过交叉注意力机制将图像特征与文本嵌入进行融合。这样以来，去噪过程不仅依赖于图像本身的特征，还结合了文本嵌入的信息。

使用训练好的模型生成图片的代码见 my_test.py, 生成的效果并没有 Stable Diffusion / Infinity 那样的好, 使用文本指令 a girl in pink/black/green dress 生成了下面三张图

图 8: 我们研制的文本条件 diffusion 生成效果

虽然没有生成一个可以辨认的小女孩，但是多少还是学到了颜色的语义信息。

2.4 Latent Diffusion

直接在像素空间中训练 Diffusion 模型对算力要求极高，且训练和推理时间极长。因此，我们参考了 Latent Diffusion Model[9] 提出的方法，不直接在像素空间训练 Diffusion 模型，而在一个压缩了的图像空间进行训练。具体实现过程如图9所示：

1. 我们先在 Flickr8k 数据集上训练一个 VQ-VAE[6] 重构图像，使用感知损失 (Perceptual Loss) 和判别损失 (Discriminative Loss) 使自编码器学习到良好的潜空间表示 (Latent Space)。
2. 使用 CLIP 的文本编码器获得文本嵌入。
3. 冻结住自编码器，在潜空间上训练一个以 U-Net 为骨架，加入交叉注意力模块的 Diffusion 模型。通过交叉注意力机制使得文本嵌入与潜空间上的图像表示得以交互。

实验参数上，对于 VQ-VAE 的训练，我们设置码本 (Codebook) 大小为 8192，下采样倍数为 2^3，使得 $256 \times 256 \times 3$ 的输入图片被压缩为 $32 \times 32 \times 4$ 的潜空间编码。我们在一张 NVIDIA RTX 3090 上对 VQVAE 训练了 60 个 epoch。重构图像效果如图10所示。

对于 Diffusion 模型的训练，我们设置训练时间步为 $T = 1000$，在一张 NVIDIA RTX 3090 上训练了 100 个 epoch。在推理阶段中，我们与2.3Our Diffusion保持一致，设置文本指令为”a girl in pink/black/green dress” 作为测试。

生成的图片如图11所示，受限于算力和训练时间，我们生成的图片并不如2.1Lora 微调 Stable Diffusion和2.2Infinity 理想。事实上，想要生成良好的潜空间表示，以及获得高清的生成图像，按

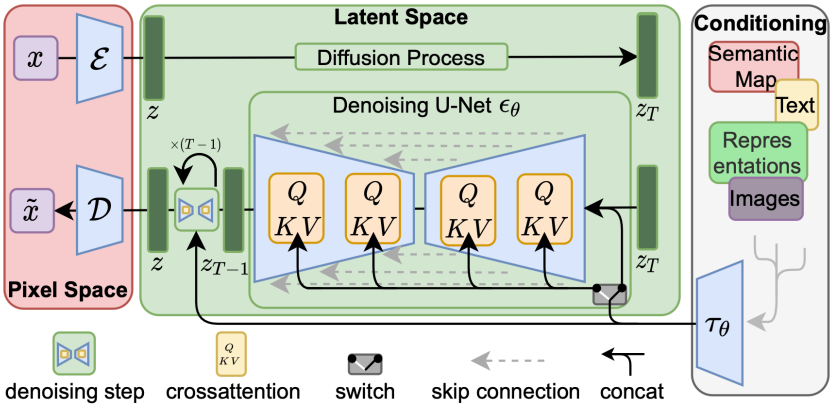


图 9: Latent Diffusion Model 训练示意图

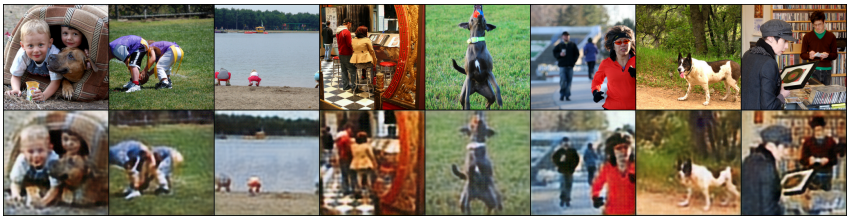


图 10: VQVAE 对 Flickr8k 中图像的重建效果

照 [9] 对 VQVAE 和 Diffusion 模型的的训练配置，我们需要在 LAION-400M 图像数据集上对 VQVAE 训练 6000 个 epoch，对 Diffusion 模型训练 1000 个 epoch。然而，从生成的图像可中可以看出，在有限的训练时间内，模型仍然学习到了一定的语义信息。我们仍旧使用使用文本指令 a girl in pink/black/green dress 生成了下面三张图：

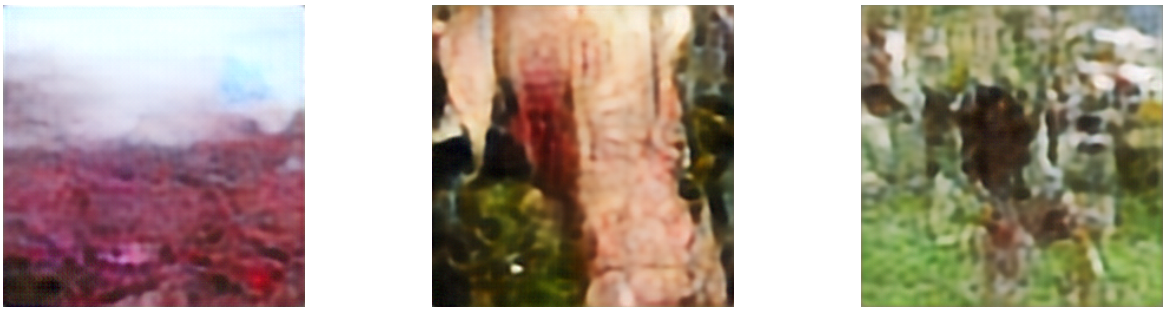


图 11: 我们训练的 ldm 生成效果

参考文献

[1] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[2] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[3] Jian Han et al. *Infinity: Scaling Bitwise AutoRegressive Modeling for High-Resolution Image Synthesis*. 2024. arXiv: 2412.04431 [cs.CV]. URL: https://arxiv.org/abs/2412.04431.

[4] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR ’16. Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90. URL: http://ieeexplore.ieee.org/document/7780459.

[5] Junnan Li et al. “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models”. In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, 23–29 Jul 2023, pp. 19730–19742. URL: https://proceedings.mlr.press/v202/li23q.html.

[6] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu koray. “Neural Discrete Representation Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

[7] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision”. In: *CoRR* abs/2103.00020 (2021). arXiv: 2103.00020. URL: https://arxiv.org/abs/2103.00020.

[8] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2023. arXiv: 1910.10683 [cs.LG]. URL: https://arxiv.org/abs/1910.10683.

[9] Robin Rombach et al. “High-Resolution Image Synthesis With Latent Diffusion Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 10684–10695.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation.” In: *CoRR* abs/1505.04597 (2015). URL: http://dblp.uni-trier.de/db/journals/corr/corr1505.html#RonnebergerFB15.

[11] Xiaolong Wang et al. “Non-Local Neural Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[12] Xi Wei et al. “Multi-Modality Cross Attention Network for Image and Sentence Matching”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[13] Peiyuan Zhang et al. *TinyLlama: An Open-Source Small Language Model*. 2024. arXiv: 2401.02385 [cs.CL]. URL: https://arxiv.org/abs/2401.02385.

附录

A 分工情况

龚舒凯

- 实现了 ResNet+Transformer Decoder 和 CLIP+Transformer Decoder 的图生文模型。
- 微调 Qformer，实现了使用 Qformer 和 TinyLlama Decoder 的图生文模型。
- 在 Flickr8k 数据集上训练 VQVAE 用于 Latent Diffusion Model 的训练

徐少轩

- 微调 Qformer，实现了基于 Qformer 的文生图模型。
- 训练并评估了 Transformer Decoder 的模型。
- 在 Flickr8k 数据集上训练 Latent Diffusion Model 。

赵嘉浩

- Stable Diffusion 在 Flickr8k 上的微调，使用预训练和微调后的版本分别生成图片。
- Infinity 的复现，生成图片。
- Our Diffusion 的实现，训练以及测试。