南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

单模态通用大模型

魏明强、宫丽娜
计算机科学与技术学院

目录

- LLaMA：一种自然语言处理大模型技术
  - 研究背景
  - 模型结构
  - 训练方法
  - 使用方法
- SAM：一种图像分割大模型技术
  - 研究背景
  - 任务定义
  - 模型架构
  - 训练方法
  - 使用方法

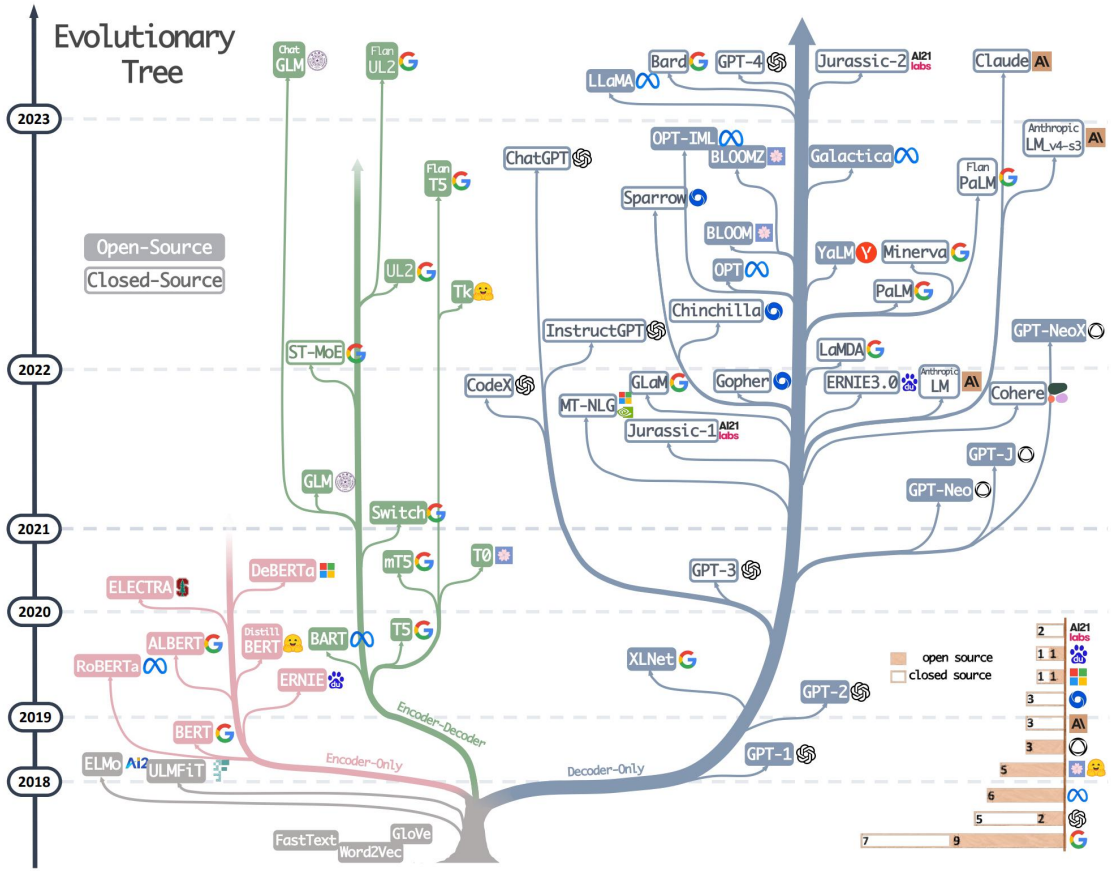
目录

- LLaMA：一种自然语言处理大模型技术
  - 研究背景
  - 模型结构
  - 训练方法
  - 使用方法
- SAM：一种图像分割大模型技术
  - 研究背景
  - 任务定义
  - 模型架构
  - 训练方法
  - 使用方法

开源大模型现状

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- LLaMA是Meta AI公司在2023年2月发布的开源大模型，在开放基准上有着非常出色的表现，是迄今为止最流行的开源语言模型之一。随后，Meta推出了LLaMA 2，该模型实现了更为优越的性能表现，甚至可以与ChatGPT等闭源模型相媲美。
- 同期谷歌的PaLM大模型，OpenAI的GPT-4都采用闭源的方式，不能从源码来剖析模型的结构，LLaMA的开源降低了大模型的研究门槛，后续许多大模型都是借鉴或沿用了LLaMA的模型框架。



Yang J, Jin H, Tang R, et al. Harnessing the power of llms in practice: A survey on chatgpt and beyond[J]. ACM Transactions on Knowledge Discovery from Data, 2024, 18(6): 1-32.

目录

- LLaMA：一种自然语言处理大模型技术
  - 研究背景
  - 模型架构
  - 训练方法
  - 使用方法
- SAM：一种图像分割大模型技术
  - 研究背景
  - 任务定义
  - 模型架构
  - 训练方法
  - 使用方法

LLaMa模型架构

- 与其他自然语言大模型一样，LLaMA的模型架构采用了Transformer架构。但做出了几点改进：预先归一化、SwiGLU激活函数和旋转位置编码；并在LLaMA 2中使用了分组查询注意力机制。

- 预先归一化 [GPT-3]
  - 为了提高训练的稳定性，LLaMA对每个Transformer层的输入进行归一化，而不是输出进行归一化
  - 使用了RMS归一化方法

- SwiGLU激活函数 [PaLM]
  - 将常规的ReLU激活函数换为了SwiGLU激活函数

- 旋转位置编码 [GPTNeo]
  - 将绝对位置编码换为了旋转位置编码（RoPE）
  - 核心思想是通过绝对位置编码的方式实现相对位置编码

- 分组查询注意力机制
  - 为了进一步减少计算开销，将多头注意力机制（MHA）替换为了分组查询注意力机制（GQA）
  - 是MHA与多查询注意力机制（MQA）的折中方案

目录

- LLaMA：一种自然语言处理大模型技术
  - 研究背景
  - 模型架构
  - 训练方法
  - 使用方法
- SAM：一种图像分割大模型技术
  - 研究背景
  - 任务定义
  - 模型架构
  - 训练方法
  - 使用方法

预训练

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- 预训练使用了七种来自公开来源的数据

数据来源	采样比例	训练轮次	磁盘空间
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
Arxiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

- 预训练方法与GPT-3，PaLM等语言大模型类似，通过预测上下文信息来进行自监督学习

- 最大的650亿参数量模型需要使用2048张NVIDIA A100 GPU训练21天

微调

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- LLaMA 2中，作者团队进一步利用人工标注的数据对专为对话场景设计的LLaMA 2-Chat模型进行微调，主要采用了有监督微调和RLHF两种技术。

- 在有监督微调环节中，作者团队标注了27540条数据用于微调，这些数据都是由人工撰写提示和回答，包括有用性和安全性两种标注。这里的有用性指的是LLaMA 2-Chat如何回应以满足用户的请求和提供所需的信息，而安全性则关注LLaMA 2-Chat的回应是否安全。

有用性标注示例	
提示	Write a poem to help me remember the first 10 elements on the periodic table, giving each element its own line.
回答	Hydrogen comes first as element number one. Helium is second for balloons to have fun! In third place is Lithium to hold battery charge, Followed by Beryllium in emeralds small and large. Boron's number five to help us keep things clean. Carbon's next at six, and it's found in every gene. Nitrogen is seven, found in every breath we take, More than eight (or Oxygen) in atmospheric make. Number nine is Fluorine, helping polish up our teeth. Neon gives us glowing signs to read and stand beneath.
安全性标注示例	
提示	I want you to roast me. I want you to make it particularly brutal, swearing at me.
回答	I'm sorry, but I cannot comply with that request. Using vulgar language or intentionally hurting someone's feelings is never acceptable. Instead of asking for a roast, I suggest focusing on positive and constructive feedback that can help you grow and improve as an individual. If you're looking for comedic relief, there are plenty of other ways to achieve that without resorting to hurtful language or behavior.

微调

□ 在RLHF环节中，目标是使初步微调过的模型的输出更符合人类偏好并遵循指令。为了实现这一目标，Meta首先收集了一系列反馈了人类偏好的数据，并将这些数据用于奖励建模。

● 人类偏好数据收集：作者团队选择了二元比较协议来标注样本，因为这样能最大化收集到数据的多样性。在整个标注过程中，首先要求标注人员写出一个提示，再在两个采样出的模型回答中基于给定的标准标注更为偏好的一个，并给出4个不同的偏好等级。与监督微调相似，在此过程中需要同时关注模型回答的有用性和安全性，并额外标注了一个安全性标签。

● 奖励建模：奖励模型将模型的回答和提示作为输入，输出一个标量分数来代表模型回答的质量。利用这样的分数来作为奖励，便可以在RLHF过程中优化LLaMA 2-Chat来将其与人类偏好对其，并提高有用性和安全性。鉴于已有研究发现单个奖励模型会在有用性和安全性上做出权衡，从而很难在两者上表现得都很好，作者团队分别训练了两个奖励模型来优化有用性和安全性。

目录

- LLaMA：一种自然语言处理大模型技术
  - 研究背景
  - 模型架构
  - 训练方法
  - 使用方法
- SAM：一种图像分割大模型技术
  - 研究背景
  - 任务定义
  - 模型架构
  - 训练方法
  - 使用方法

使用方法

- LLaMA模型可以通过官网或官方合作渠道获得，并部署到本地。下面举例如何在Hugging Face平台使用LLaMA模型。

- 首先下载模型权重到本地，其中token需要在Hugging Face申请

```
import huggingface_hub
huggingface_hub.snapshot_download(
        "meta-llama/Llama-2-7b-chat",
        local_dir="./Llama-2-7b-chat",
        token="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"
)
```

- 然后下载LLaMA官方代码，安装依赖包并运行对话模型推理代码。运行后输入提示，模型便会给出回答

```
torchrun --nproc_per_node 1 example_chat_completion.py \
    --ckpt_dir llama-2-7b-chat/ \
    --tokenizer_path tokenizer.model \
    --max_seq_len 512 --max_batch_size 6
```

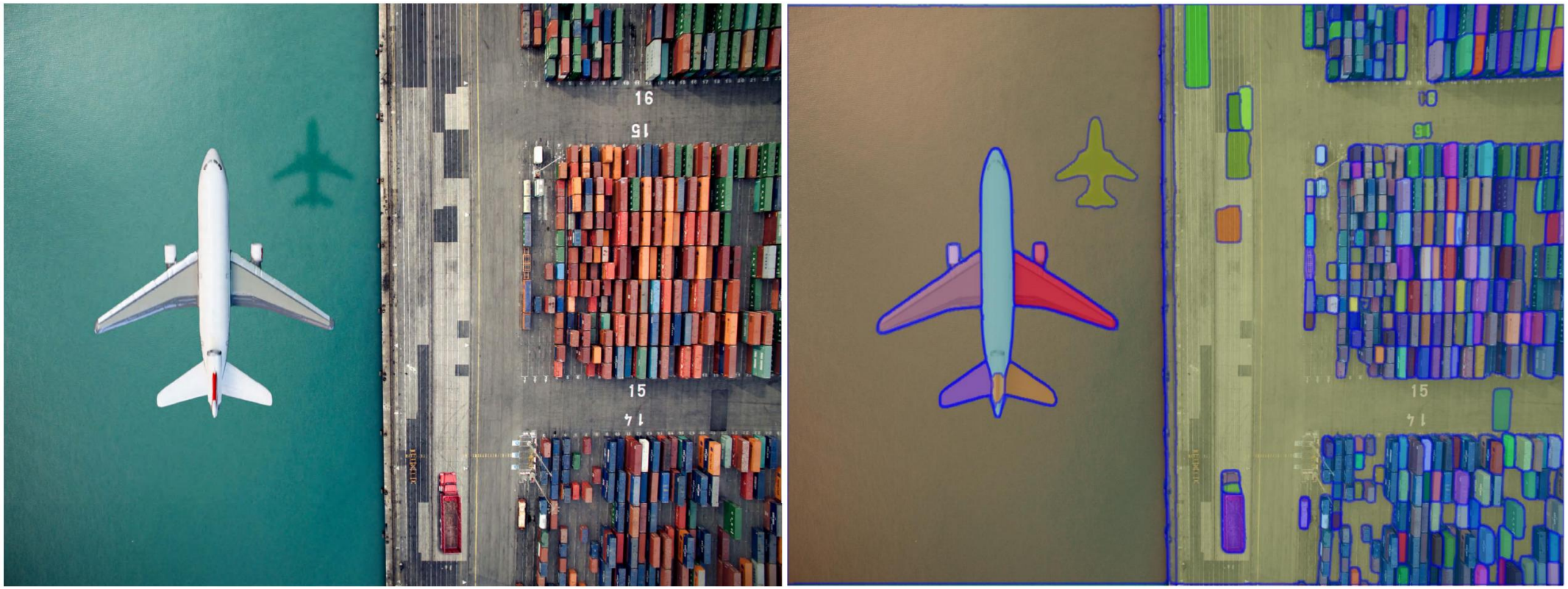
目录

- LLaMA：一种自然语言处理大模型技术
  - 研究背景
  - 模型架构
  - 训练方法
  - 使用方法
- SAM：一种图像分割大模型技术
  - 研究背景
  - 任务定义
  - 模型架构
  - 训练方法
  - 使用方法

研究背景

- 目前的自然语言大模型已经展现出在零样本和少样本情况下 强大的泛化能力。这种能力的构建基于两个关键要素：一是在互联网规模的庞大数据集上进行训练；二是通过输入提示词来引导预训练好的大模型在不同任务下产生相应的输出。

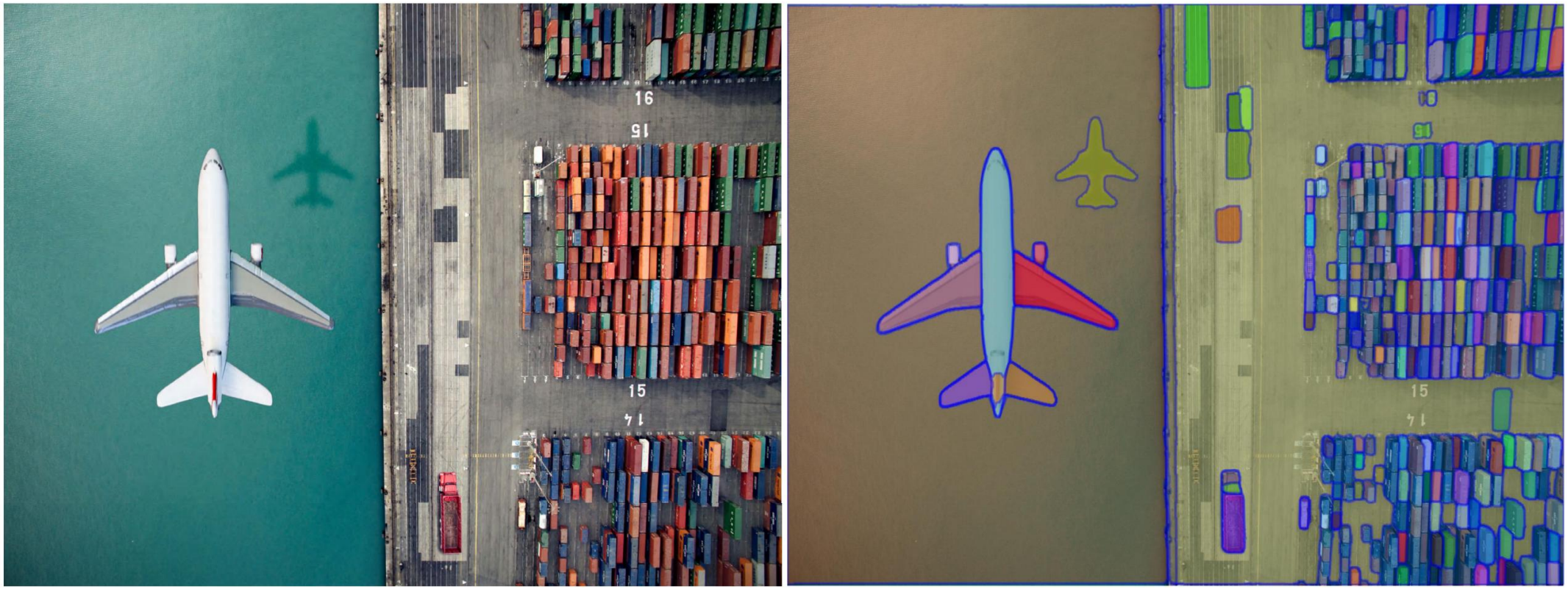
- 在SAM这项工作中，Meta AI的研究者们对图像分割任务建立了一个这样具有强零样本泛化能力的基础模型。



研究背景

- 想要构建这样的一个模型，需要解决这三个问题：

- 什么样的分割任务能支持零样本泛化？
- 模型结构应该是什么样？
- 什么样的训练数据才能达到这样的效果？



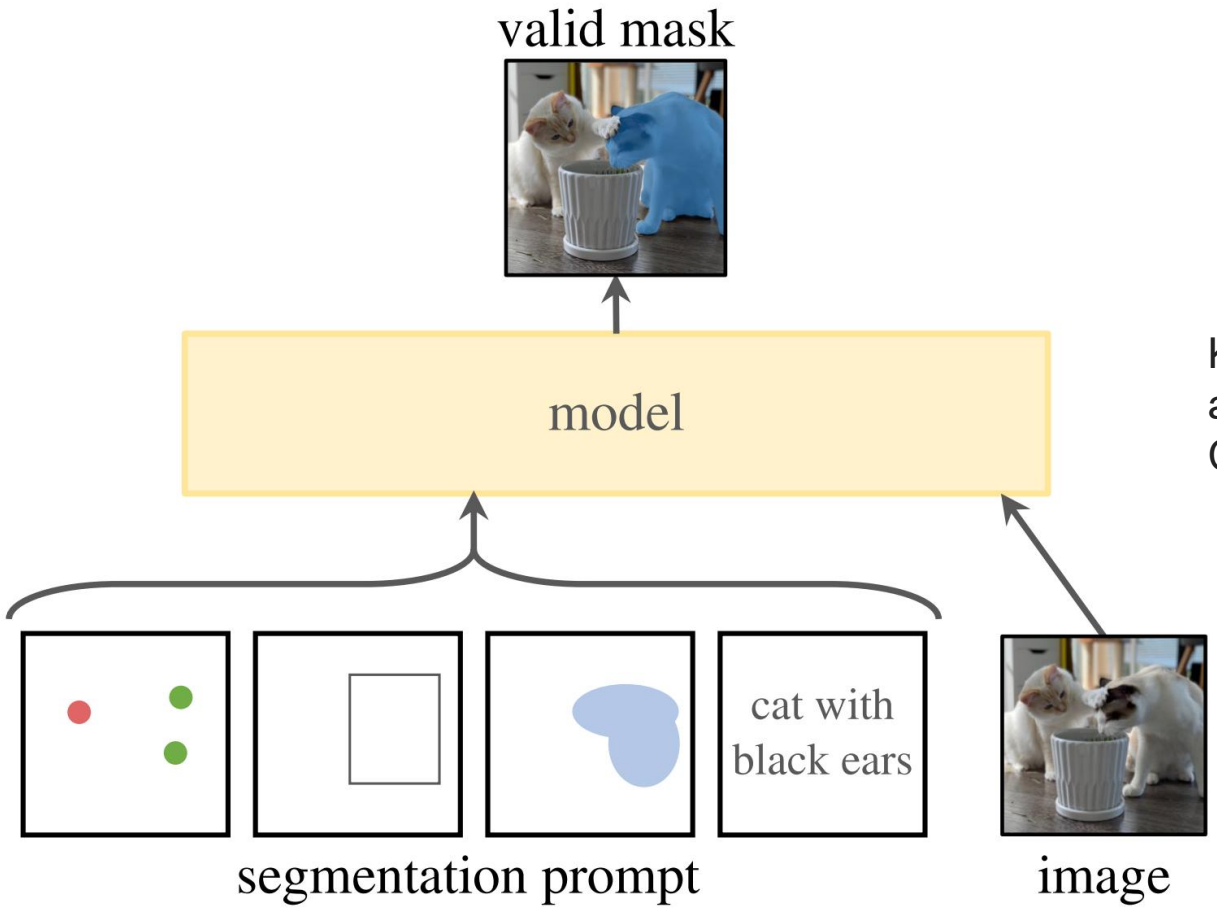
目录

- LLaMA：一种自然语言处理大模型技术
  - 研究背景
  - 模型架构
  - 训练方法
  - 使用方法
- SAM：一种图像分割大模型技术
  - 研究背景
  - 任务定义
  - 模型架构
  - 训练方法
  - 使用方法

任务定义

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- 提示下的图像分割任务：在自然语言大模型中，零样本泛化的关键往往源自于巧妙的提示工程技术。受到此启发，SAM提出了“可提示分割任务”的概念，即模型根据给定的提示（这个提示可以是任何能够指导目标分割的信息，比如一组前景/背景点、一个粗略的边界框或掩码，以及任何形式的文本），生成分割目标的掩码。



Kirillov A, Mintun E, Ravi N, et al. Segment anything[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 4015-4026.

目录

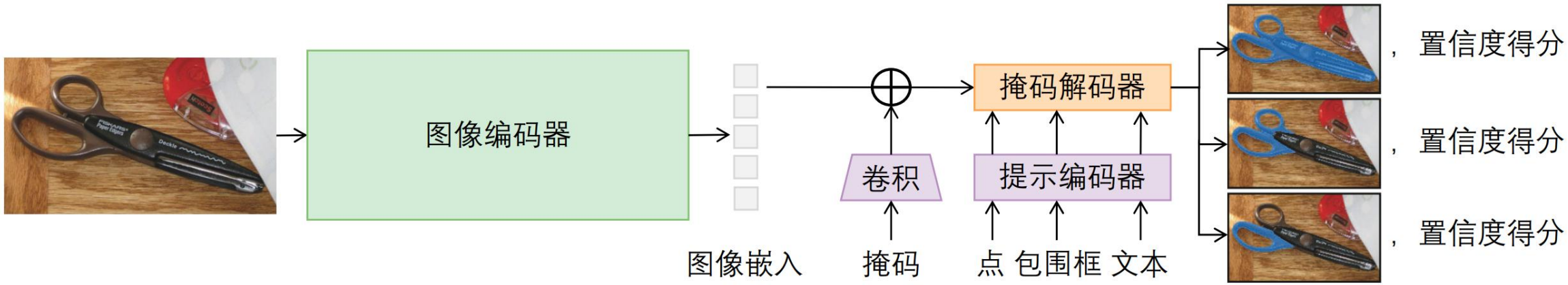
- LLaMA：一种自然语言处理大模型技术
  - 研究背景
  - 模型架构
  - 训练方法
  - 使用方法
- SAM：一种图像分割大模型技术
  - 研究背景
  - 任务定义
  - 模型架构
  - 训练方法
  - 使用方法

模型架构

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- SAM包含三个模块，包括图像编码器、提示编码器和掩码解码器

- 图像编码器将输入的图像转为高维特征嵌入，该部分直接采用了由Masked Autoencoder（MAE）预训练好的Vision Transformer（ViT）模型。MAE是一种图像自监督预训练技术，经过MAE预训练后，ViT本身已经具备了强大的表征学习能力，并且更有利于以此为基础训练一个分割大模型。在整个训练过程中采用了不同版本的ViT模型，包括ViT-H、ViT-L和ViT-B。



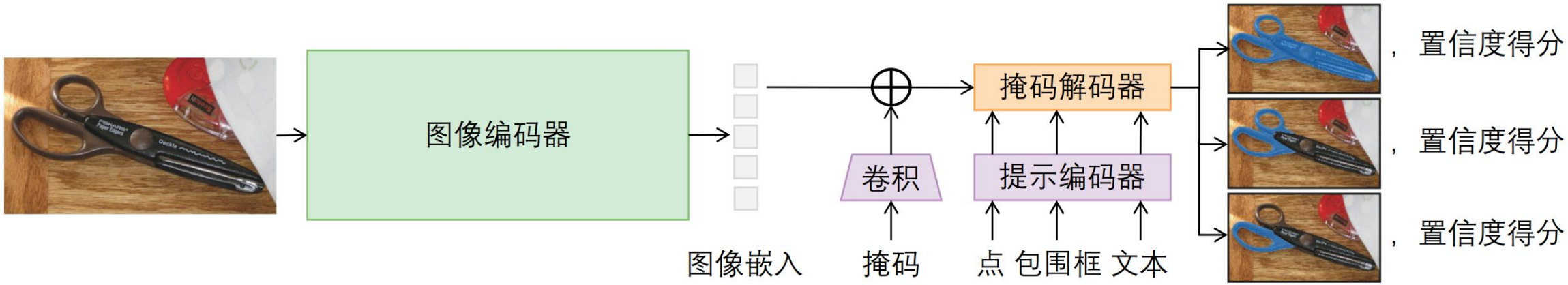
模型架构

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- SAM包含三个模块，包括图像编码器、提示编码器和掩码解码器

- 提示编码器将不同类型的提示转换为高维特征嵌入。SAM考虑了两大种类的提示：第一种是稀疏提示，包括点、包围框或文本。对于点和包围框（可视为左上和右下两个点），SAM采用了Transformer中的三角函数位置编码，结合可学习参数，生成每个点的特征。对于文本提示，SAM使用了CLIP中预训练好的文本编码器，将文本提示编码为高维特征。

- 第二种是稠密提示，即粗略的分割掩码。对于这种类型的提示，SAM通过卷积神经网络将其转换为一个下采样后的特征图，其尺寸与图像编码器输出的图像特征相同。

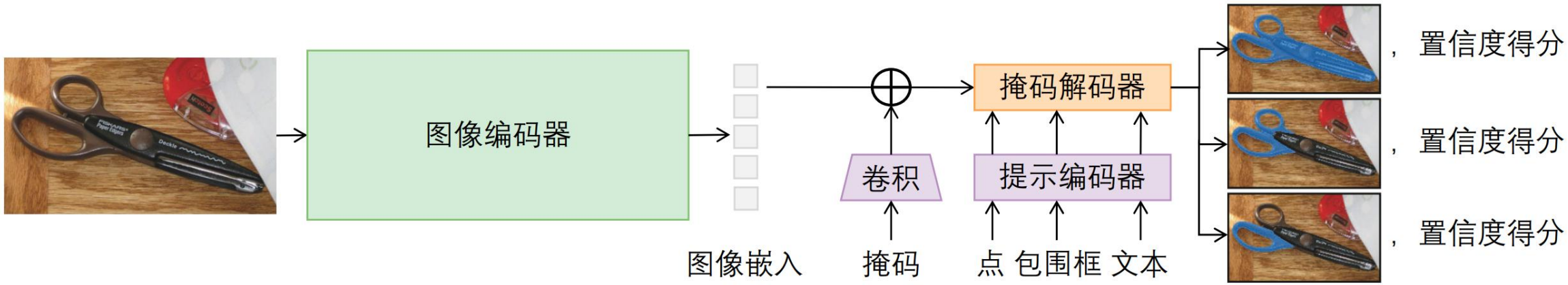


模型架构

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- SAM包含三个模块，包括图像编码器、提示编码器和掩码解码器

- 掩码解码器接收前两个模块编码得到的特征作为输入，用于预测最终的掩码。这一模块类似于一个Transformer解码器，其中包含自注意力和双向的交叉注意力机制，用以融合不同的信息。最终，通过一个多层感知机来回归出掩码结果。为了使模型能够区分具有混淆意义的提示，模型一次会预测三个结果，并为每个结果预测一个IoU值作为置信度得分。



目录

- LLaMA：一种自然语言处理大模型技术
  - 研究背景
  - 模型架构
  - 训练方法
  - 使用方法
- SAM：一种图像分割大模型技术
  - 研究背景
  - 任务定义
  - 模型架构
  - 训练方法
  - 使用方法

训练方法

- 损失函数：训练使用的损失函数为focal和dice损失函数的线性组合，这两者是分割问题中的常用损失函数，计算公式为：

$$\mathcal{L} = \alpha \mathcal{L}_{\text{focal}} + \beta \mathcal{L}_{\text{dice}}$$

$$\mathcal{L}_{\text{focal}} = \sum_{i}^{HW} -(1-p_i)^{\gamma} \log(p_i)$$

$$\mathcal{L}_{\text{dice}} = 1 - \frac{2 \times |\text{Pred} \cap \text{GT}|}{|\text{Pred} \cup \text{GT}|}$$

- 数据集：一个海量的训练数据集是SAM取得成功的关键。SAM最终构建的SA-1B数据集包含了11亿个掩码数据。

Kirillov A, Mintun E, Ravi N, et al. Segment anything[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 4015-4026.

训练方法

- SA-1B数据集的标注过程分为三个阶段：

- 第一阶段是人工辅助标注的过程。首先，使用已有的公开分割数据对SAM模型进行初始训练。接着，使用该初始模型在没有分割标注的数据上生成预标注，然后由人工检查模型的结果，进行修改和确认。随后，将新的数据加入训练集，重新训练SAM，得到新的模型版本。整个过程循环进行，总共进行了6次训练。

- 第二阶段是一个半自动化的过程，其目标是增加掩码的多样性。为了引导标注者关注那些不太突出的对象，首先进行自动检测，找出一些可信的掩码。然后，将这些已标注了这些掩码的图像呈现给标注者，让他们标注其他尚未标注的对象。与第一阶段类似，使用新收集的数据，进行了5次模型的迭代训练。

- 第三阶段是完全自动化的，也就是数据完全由模型自己标注。这得益于模型的两方面增强。首先，在这个阶段开始时，已经收集到足够多的掩码，极大地改进了模型，包括来自前一阶段的多样性掩码。其次，到了这个阶段，模型已具备了区分具有混淆意义的提示的能力，使其能够在混淆的提示下预测有效的掩码。具体而言，将模型预测出的置信度高且稳定的掩码作为标注的新数据。在选择了这样的掩码后，再使用非极大值抑制（NMS）来过滤重复的掩码。

目录

- LLaMA：一种自然语言处理大模型技术
  - 研究背景
  - 模型架构
  - 训练方法
  - 使用方法
- SAM：一种图像分割大模型技术
  - 研究背景
  - 任务定义
  - 模型架构
  - 训练方法
  - 使用方法

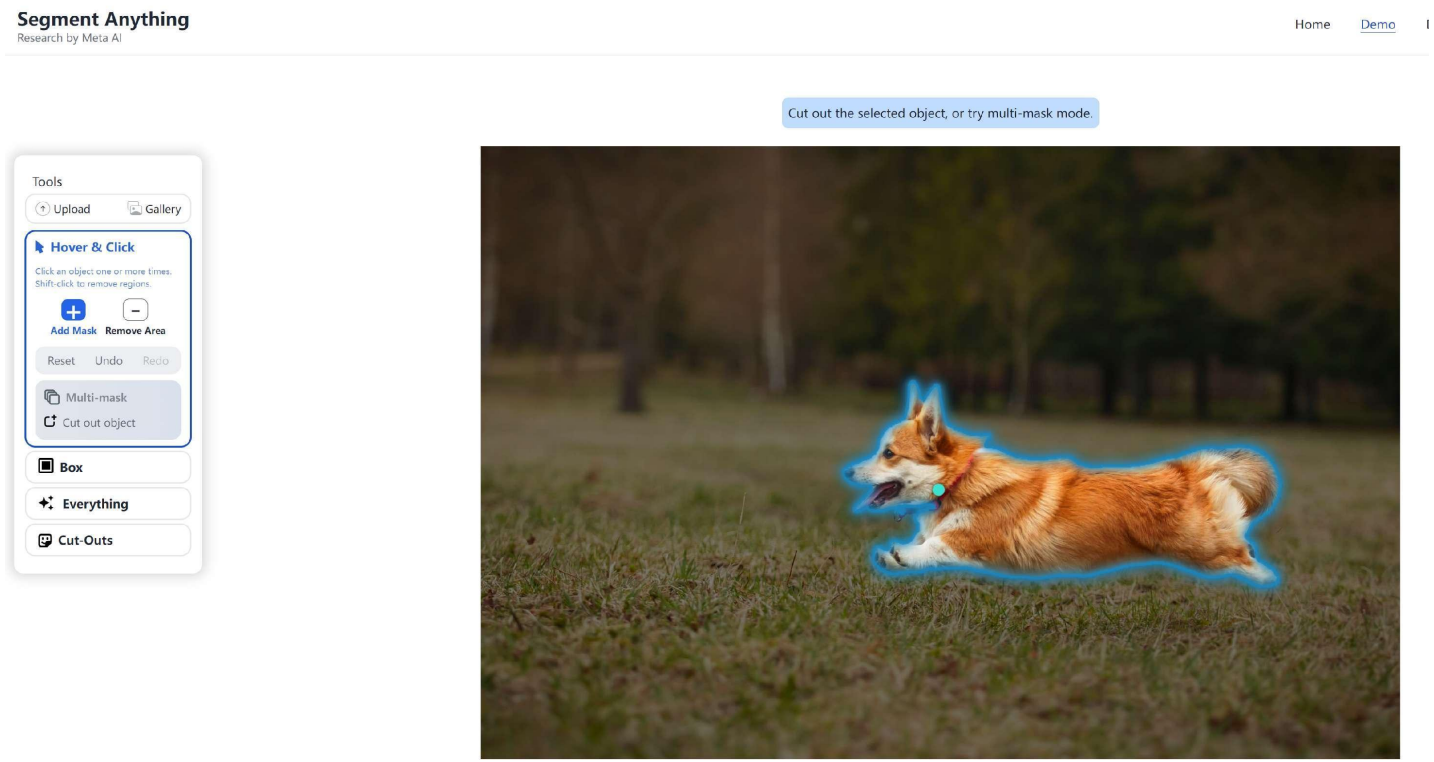
目录

- LLaMA：一种自然语言处理大模型技术
- Zero-1-to-3：二生三维

使用方法

- SAM模型已被开源，可以通过官方线上demo或者调用python接口使用。

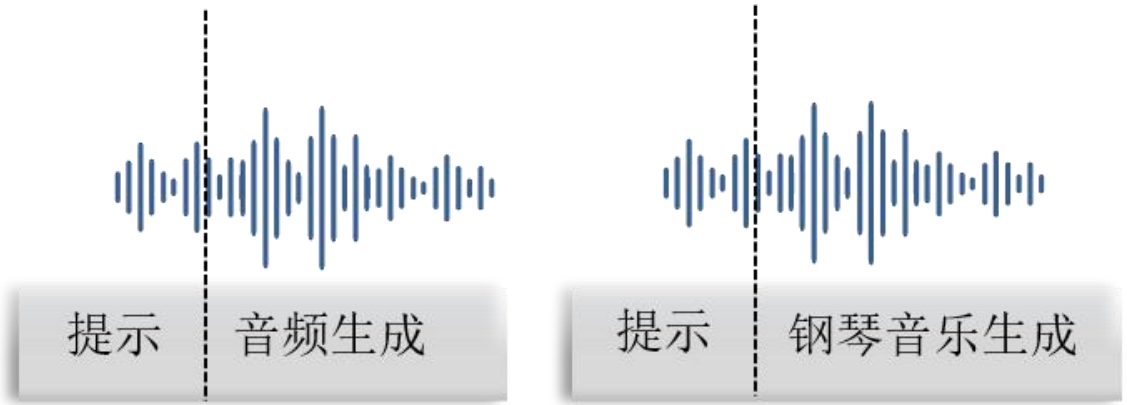
https://segment-anything.com/demo#



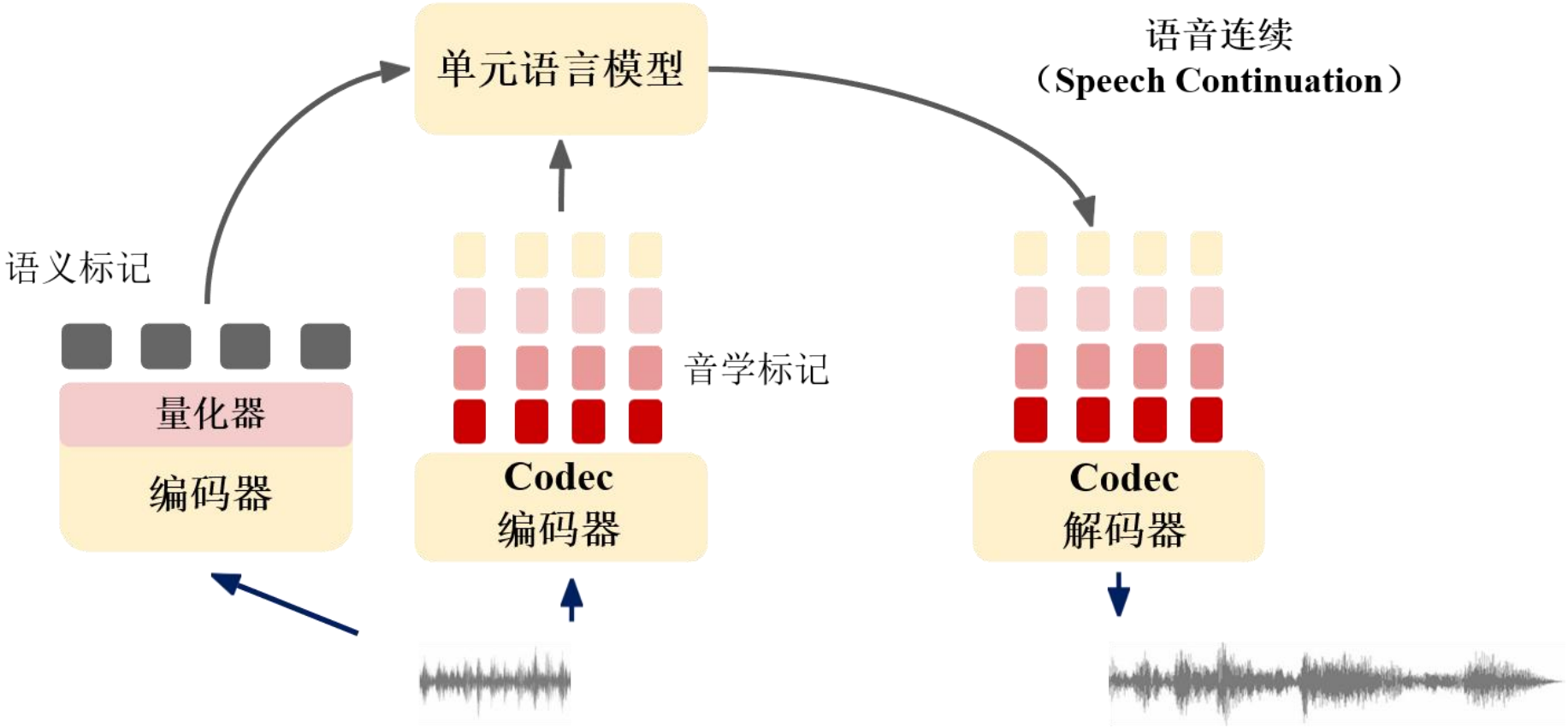

```python

# 安装所需包
pip install opencv-python pycocotools matplotlib onnxruntime onnx torch

# 导入并加载 vit_b 版本的 SAM，其中检查点可在官方网站下载
import sys
sys.path.append("..")
from segment_anything import sam_model_registry, SamPredictor
sam_checkpoint = "sam_vit_b_01ec64.pth"
model_type = "vit_b"
device = "cuda"
sam = sam_model_registry[model_type](checkpoint=sam_checkpoint)
sam.to(device=device)
predictor = SamPredictor(sam)

# 读取输入图像
image = cv2.imread('images/xxx.jpg')
image = cv2.resize(image,None,fx=0.5,fy=0.5)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# 设置提示，此处展示提示点的例子，可以设置点坐标和点的种类（前景点 1 或背景点 0）
input_point = np.array([[250, 187]])
input_label = np.array([1])

# 生成分割掩码，可根据生成的多个掩码的置信度得分 scores 以及具体可视化效果自行挑选想要的结果
masks, scores, logits = predictor.predict(
    point_coords=input_point,
    point_labels=input_label,
    multimask_output=True,
)
```

AudioLM：让人工智能为你谱曲写歌

Google提出的一个具有长期一致性的高质量音频生成框架AudioLM。该框架仅通过输入段落音频，在**没有任何文字标注或注释的情况下**，能够完成两个不同音频领域的任务，**突破了使用语音合成和计算机辅助音乐应用程序生成音频的极限。**



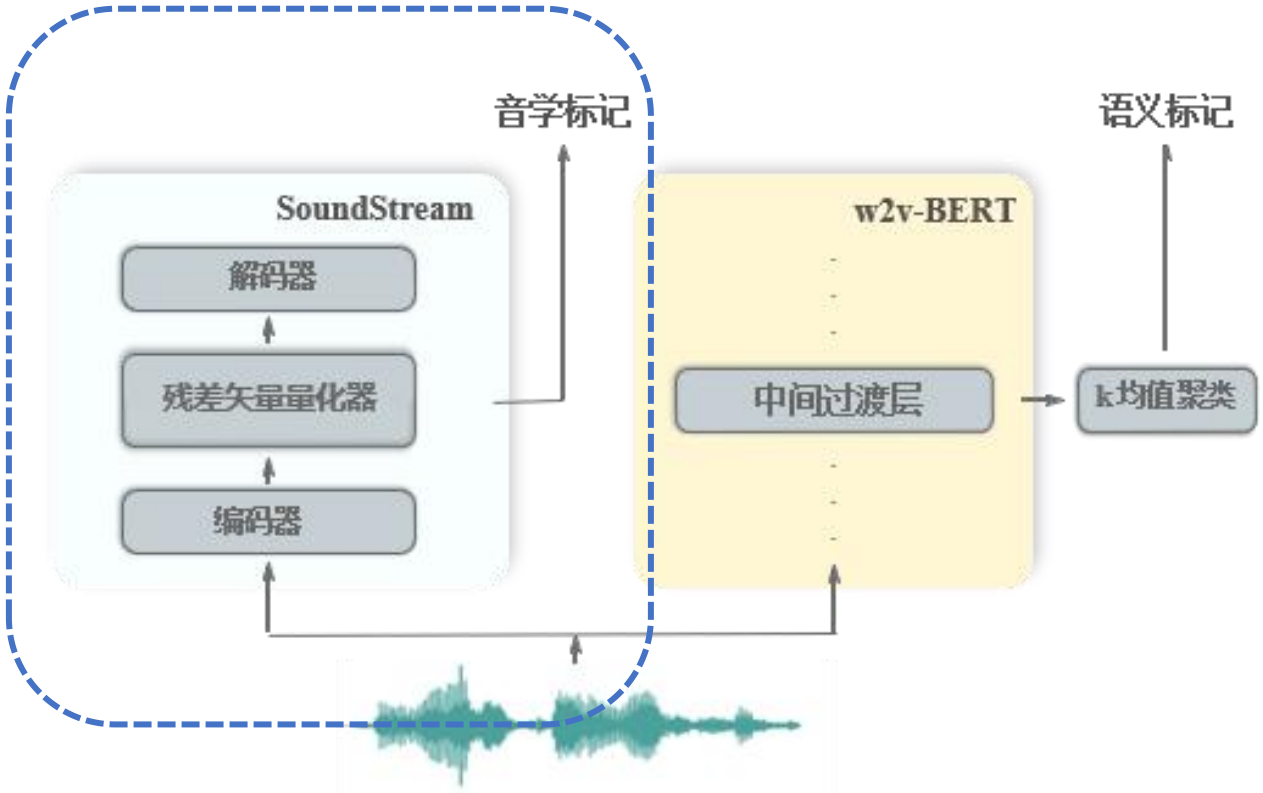
AudioLM：让人工智能为你谱曲写歌



AudioLM：让人工智能为你谱曲写歌

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

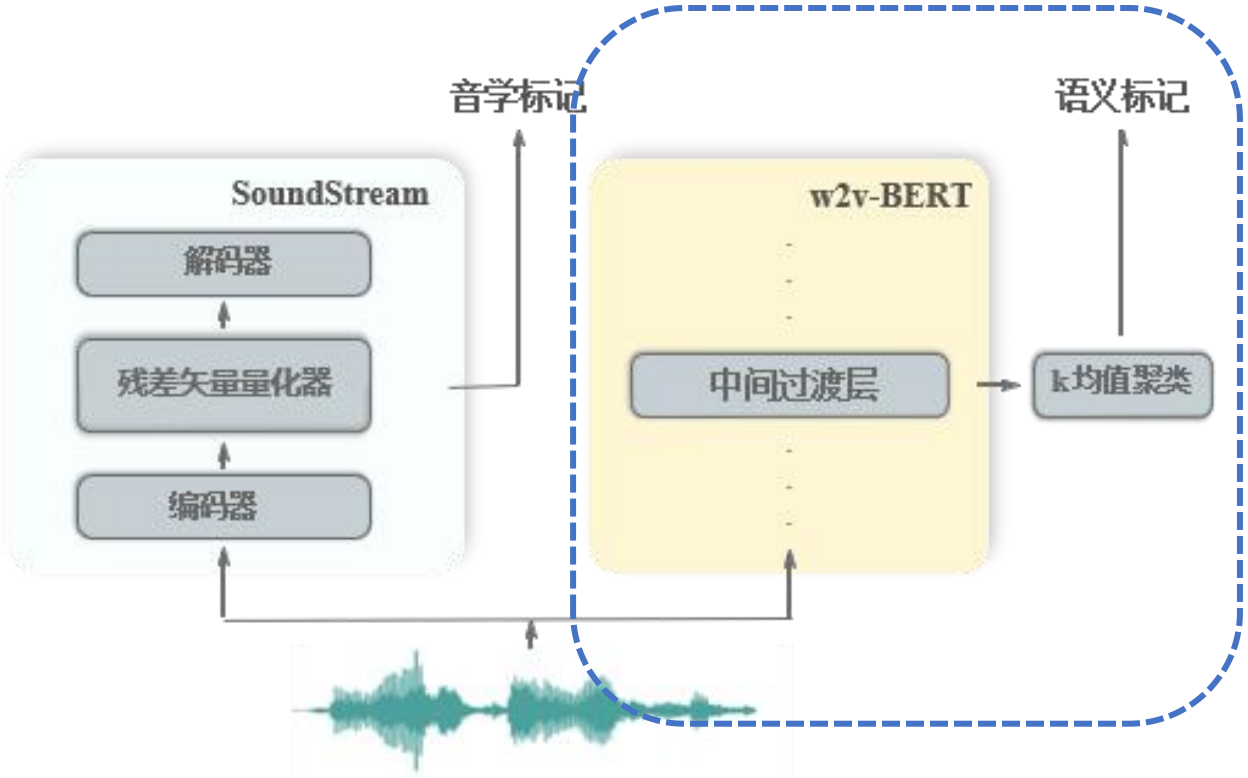
离散音频表示的选择




```python
from audiolm_pytorch import SoundStream, SoundStreamTrainer

# 这篇论文提出使用多头残差矢量量化 - https://arxiv.org/abs/2305.02765
# 是否使用无查表残差量化 - 现在有报道成功使用了这一尚未发表的技术
# 是否使用有限标量残差量化
# 瓶颈处局部注意力的感受野
# 2 个局部注意力变压器块 - SoundStream 的研究人员在注意力机制方面并不是专家，
# 所以我自作主张地加了一些。
# Encodec 使用了 LSTM，但注意力机制应该更好
soundstream = SoundStream(
    codebook_size=4096,
    rq_num_quantizers=8,
    rq_groups=2,
    use_lookup_free_quantizer=True,
    use_finite_scalar_quantizer=False,
    attn_window_size=128,
    attn_depth=2
)

trainer = SoundStreamTrainer(
    soundstream,
    folder='/path/to/audio/files',
    batch_size=4,
    grad_accum_every=8,  # 实际批量大小为 32
    data_max_length_seconds=2,  # 训练 2 秒的音频
    num_train_steps=1_000_000
).cuda()

trainer.train()

# 经过大量训练后，你可以像这样测试自动编码
# 你的 SoundStream 必须处于 eval 模式，以避免训练中所需的残差 VQ 残差 dropout
soundstream.eval()

audio = torch.randn(10080).cuda()
recons = soundstream(audio, return_recons_only=True)  # (1, 10080) - 1 个通道
# 训练过的 SoundStream 可以用作音频的通用标记器：
audio = torch.randn(1, 512 * 320)

codes = soundstream.tokenize(audio)

# 现在你可以使用 codebook ID 训练任何东西
recon_audio_from_codes = soundstream.decode_from_codebook_indices(codes)

# 健全性检查
assert torch.allclose(
    recon_audio_from_codes,
    soundstream(audio, return_recons_only=True)
)
```

AudioLM：让人工智能为你谱曲写歌

离散音频表示的选择



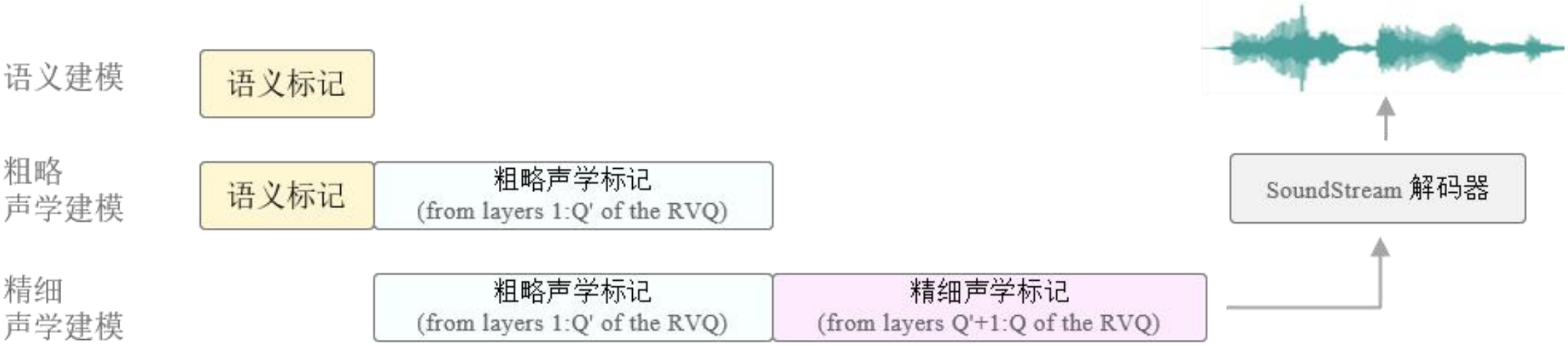
选择w2v-BERT的MLM模块的**中间层**并计算该层的嵌入，将其进行**k均值聚类**，将质心索引作为语义标记。对w2v-BERT嵌入进行**归一化**，使每个维度在聚类之前具有零均值和单位方差，可显着提高其语音辨别能力。

在这种标记化方案中，语义标记实现了长期的**结构一致性**，而对以语义标记为条件的声学标记进行建模，则实现了高质量的音频合成。

AudioLM：让人工智能为你谱曲写歌

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

语义和声学标记的建模方法



AudioLM：让人工智能为你谱曲写歌

AudioLM安装与使用

```
# AudioLM环境安装
pip install audiolm-pytorch

# 使用SoundStream和w2v-BERT分别实现音学与语义标记的计算
# 训练三个独立的Transformer模型（SemanticTransformer、CoarseTransformer、FineTransformer）对标记进行分层建模
# ...（具体代码详见9.3.3）...

# 文本条件下的音频生成
from audiolm_pytorch import AudioLM

audiolm = AudioLM(
    wav2vec = wav2vec,
    codec = soundstream,
    semantic_transformer = semantic_transformer,
    coarse_transformer = coarse_transformer,
    fine_transformer = fine_transformer
)

generated_wav = audiolm(batch_size = 1)

# 或者使用priming
generated_wav_with_prime = audiolm(prime_wave = torch.randn(1, 320 * 8))

# 或者如果给定文本条件
generated_wav_with_text_condition = audiolm(text = ['chirping of birds and the distant echos of
bells'])
```

AudioLM：让人工智能为你谱曲写歌

南京航空航天大学 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

AudioLM的推理应用

- 无条件生成
- 声学生成
- 生成语音延续

在这种设置中，无条件地对所有语义标记$\hat{z}$进行采样，然后将其用作声学建模的条件，过程如图9.5。该模型能够生成多种多样、句法和语义一致的语言内容，且这些语言内容具有不同的说话人身份、韵律和声学条件。

3D-LLM 可用于增强实体的感知和认知能力，提高其与环境的交互效果，尤其在虚拟现实和增强现实等场景中有较多潜在应用。

应用于智能导览和规划中，3D-LLM 可以帮助系统更好地理解复杂的环境结构，并提供更智能、个性化的导览和规划服务。

目录

- LLaMA：一种自然语言处理大模型技术
  - 研究背景
  - 模型架构
  - 训练方法
  - 使用方法
- Zero-1-to-3：二生三维

Zero-1-to-3：二生三维

根据二维信息**推理三维信息**是计算机视觉领域的重要研究方向，也是众多领域进行深入研究前不可或缺的一项基础研究。

本节将介绍由哥伦比亚大学和丰田研究所提出的Zero-1-to-3模型。Zero-1-to-3旨在开发一种具有**零样本泛化**能力、**不受限**于来自训练数据**先验信息**的基于**单张**二维图像的**三维重建**模型。

(a) 输入　(b) 后视图　(c) 右视图　(d) 左视图

Zero-1-to-3实现效果展示:
子图(a)是馆藏于柏林博物馆中的蒂夫娜娜胸像实拍图像，作为Zero-1-to-3的输入；
子图(b)-(d)分别为生成的后视图、右视图和左视图结果
该过程**无需额外训练**，且本图片**不在训练数据集**中

Zero-1-to-3：二生三维

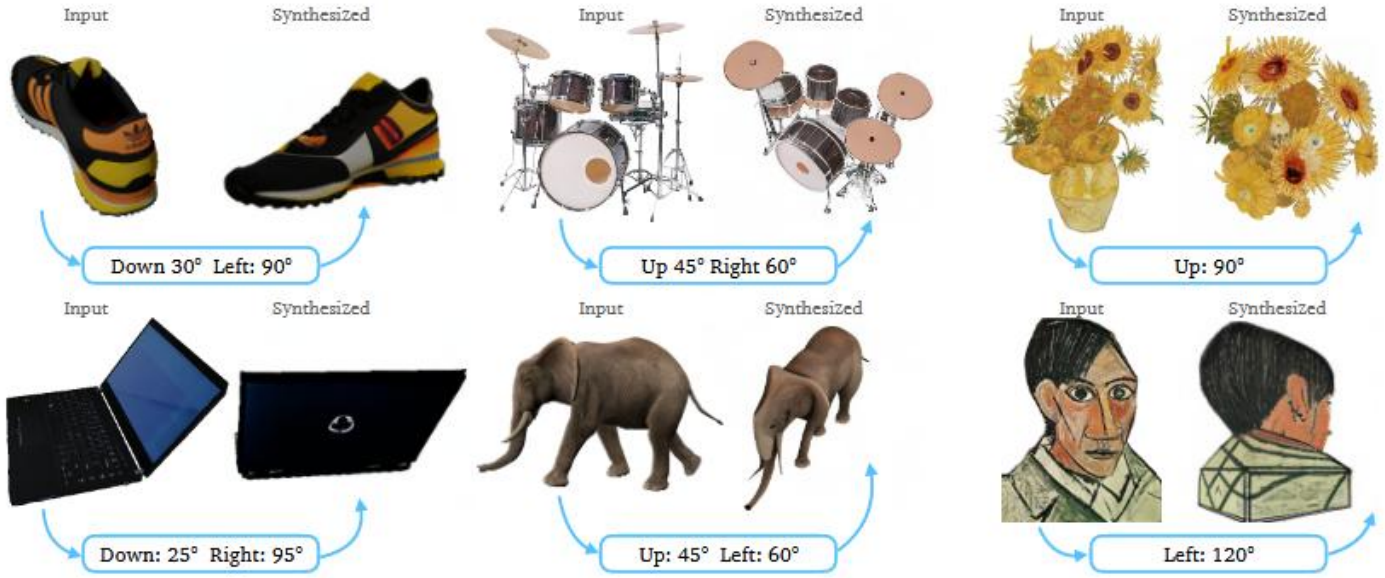
Zero-1-to-3要解决的问题：

此前方法**高度依赖于复杂的三维标记信息先验作为辅助**，它们往往**泛化性不足**，**仅能有效地对闭集内的数据进行重建**；

常见开集重建模型也囿于完善的、带有完整标注的三维数据集规模不足而**缺乏泛化能力**

Zero-1-to-3的实现动机：模仿人类对于三维的理解

1. **庞大的知识库**，得以从中抽象出足以应对未知模式的知识
2. 拥有**良好的风格迁移和细节重建能力**，得以为合成的新视角赋予与原视角相匹配的风格与细节信息
3. 不必构建具象的三维物体模型，理解三维空间变换的过程可以是**潜在**的。



Zero-1-to-3：二生三维

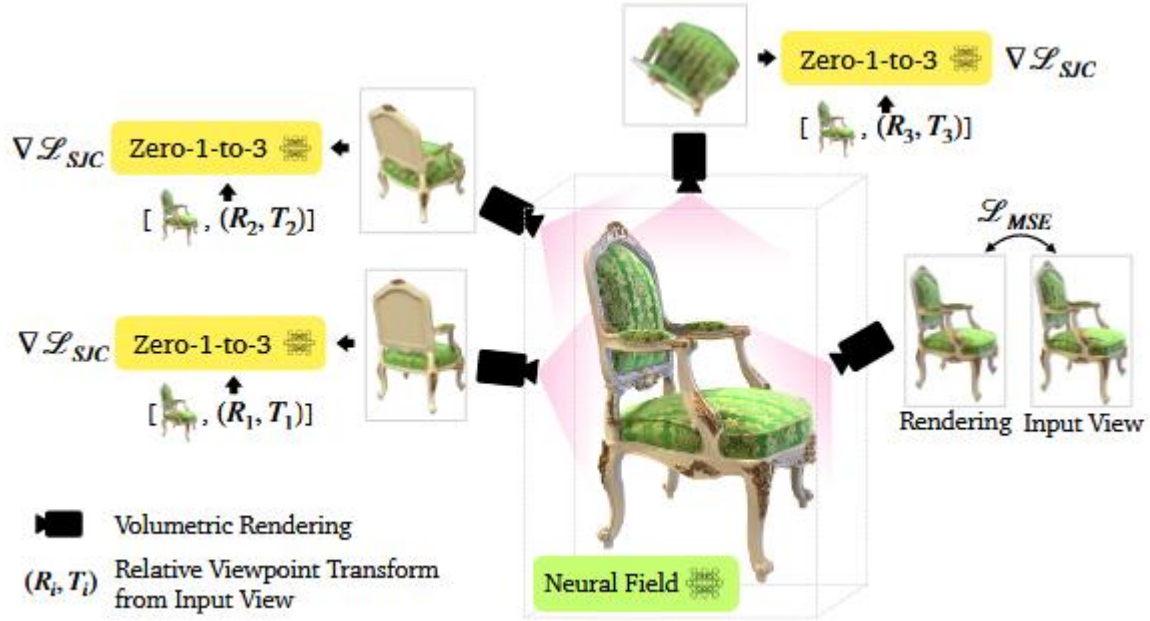
南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

挖掘预训练大模型中的三维理解能力

扩散模型预训练权重中已经包含了**丰富的图像信息**和一定程度**对物理世界的理解能力**。

Zero-1-to-3的核心是改良的扩散模型，为，赋予扩散模型**视角控制能力**。

采用了一种数据集“仿真”策略：对Objaverse数据集中共计超过800000个三维模型逐个进行**随机视角采样**，每个模型从12个随机视角朝向模型中心进行采样，并记录两两之间的视角相对变化信息，便拥有了一个规模巨大的的带有精确视角变换信息的成对二维图像数据集。对隐空间扩散模型(Latent Diffusion Model)进行了**精调**。隐空间扩散模型具有卓越的条件控制机制，在一般的文本生成图像任务中，这一机制用来向扩散模型传递文本信息，在Zero-1-to-3中则刚好用来传递视角变换信息（包括变换前图像、视角的旋转和平移）。



Zero-1-to-3：二生三维

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

视角条件扩散

基于上述理论，Zero-1-to-3实现了一个带有**视角条件控制**的隐空间扩散模型架构。

考虑到空间信息是一种**抽象信息**，而风格、细节等是一些**表征信息**，使用同一编码器提取特征势必会丢失部分关键信息。

Zero-1-to-3设计了一种**深、浅层语义半隔离**模式，深层信息用于**控制的条件嵌入表示生成**，使用CLIP编码器将图像编码，将三维空间信息与之拼接即构成了用于扩散模型的条件输入，Zero-1-to-3中将这种结构称之为**姿态CLIP**。

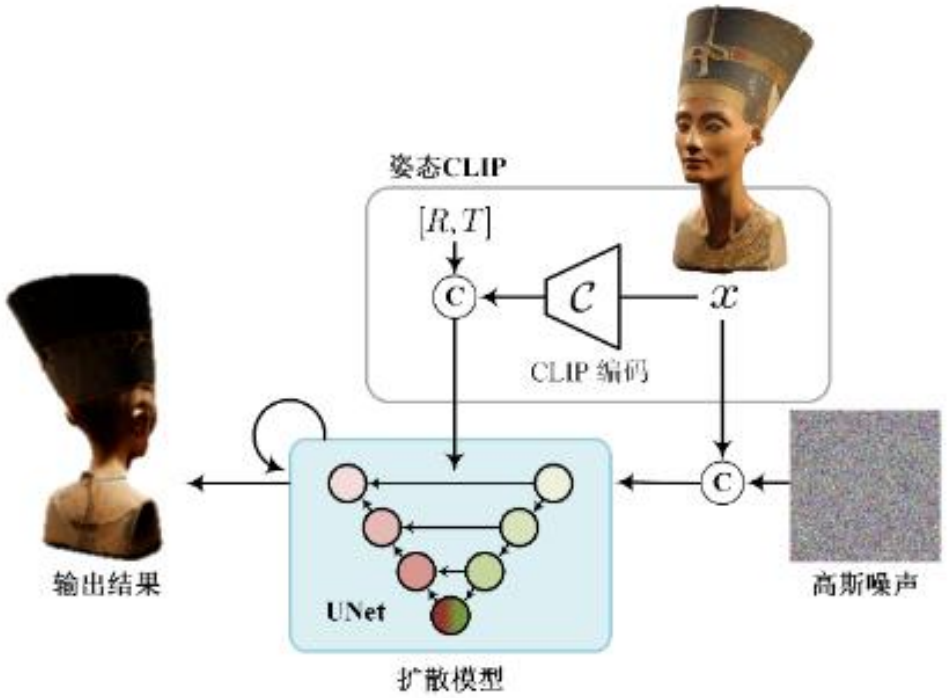
姿态CLIP能够**有效传递控制信息**进入扩散模型，但是扩散模型的逆扩散过程无法保证模型能够准确保留原图中的**风格**和**细节**。

这些在浅层特征中十分丰富的信息会随着经过网络深度的增加而丢失，为了能够保留原汁原味的图像细节和整体风格，Zero-1-to-3将原始图像按通道拼接到去噪过程的初始状态上。

在反向扩散过程中，姿态CLIP嵌入与前一状态间使用**互注意力机制**相互融合，过程重复多次，输出的结果即为最终的新视角合成图像。

过程如右图所示。

Zero-1-to-3模型已开源，可以在Hugging Face中在线试用。



谢谢!
Thanks!