点云的多模态基础模型：Openshape，UNI3D和ULIP2

Openshape

论文：**OpenShape：LIU M, SHI R, KUANG K, et al. OpenShape: Scaling Up 3D Shape Representation Towards Open-World Understanding.**

代码：https://github.com/Colin97/OpenShape_code

1.解决了什么问题？创新点?

1）OpenShape：一种学习文本、2D图像和点云（3D shape）多模态联合表示的方法。

2）问题：在解决3D形状理解任务时， 3D数据集的规模不足，并且在面对没见过的形状类别时的泛化能力不足。通常3D的任务中场景的解决方法主要是利用大规模预训练的2D图像-文本模型。而在3D形状理解的工作中，直观的想法是将3D投影到2D，并使用CLIP来分析图像。然而，这种方法存在一些限制，如遮挡、投影过程中的信息丢失、由多个CLIP推理带来的渲染延迟。因此，从预训练的2D模型中提取知识，训练一个直接利用3D数据的模型似乎更为可行。

3）作者提出通过扩展表示学习来解决这些问题，涉及四个关键因素：数据规模、文本质量、3D骨干网络的扩展以及数据重采样。作者在零样本3D形状分类任务的下游任务上做了实验。创新点：

1.在处理长尾类别方面表现卓越。2.学习了广泛的视觉和语义概念。3.可与现成的基于CLIP的模型集成。

2.用了什么模型？method?

本文的方法主要包含两个方面：数据处理（数据集集成和文本处理）和跨模态对齐（对比学习框架、扩大3D骨干网络、硬负样本挖掘）

(a) Ensemble Datasets (b) Text Filtering & Enrichment (c) Cross-Modal Alignment (d) Cross-Modal Applications

一、数据处理

1. 集成3D数据集 与 多模态对齐表示：

1.数据集：包括上图中（a）的4个数据集，其中ShapeNetCore、3D-FUTURE和ABO是经过人工验证的，质量高但是规模小。Objaverse是由网上用户上传的，规模大且更加多样化，但是质量和分布不均衡，因此需要做后续的文本处理操作。

2.定义了一个元组 $\{(P_i, T_i, I_i)\}$ ，表示含义为（P：3D点云，T：文本，I：图像）。

点云是由网格表面中采样得到的：对于每个形状，从网格表面采样10,000个点，并根据网格纹理插值点颜色；

图像是从预定义的12个相机参数渲染得到的，还可能包括缩略图，在训练时随机选择其中一张；

文本首先从objaverse的模型名、或其他数据集的metadata得到，然后进行后续的处理。

2.文本处理：

文本处理的目的是对齐3D shape和文本，具体流程可见上图的（b），具体有三个大步骤，文本过滤、图像描述和图像检索。

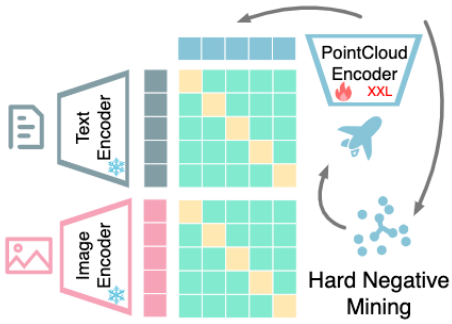
1）文本过滤 Filtering：输入为原始文本，使用 GPT-4 过滤噪声，例如时间戳、纯模型数字、难以理解的描述、随机文件名和随机字符。通过 GPT-4，GPT-4可以过滤掉了大约 30% 的原始用户文本。

2）图像描述 Captioning：使用BLIP, Azure cognition services为正面渲染的图像进行描述，得到两条描述，两条描述可以增强或替代质量低的文本，并且可以相互补充。

3）图像检索Image Retrieval：使用CLIP ViT-L检索索引从LAION-5B数据集中检索形状渲染的k-NN图像。然后将k-NN 图像的描述作为我们 3D 模型的检索文本。与图形描述模型生成相比，检索到的文本涵盖了更广泛的文本风格，更细粒度。

训练时，从中随机选择一种来源，在从中随机选择一个文本，最后做prompt engineering，即由多个模板生成句子，并取其平均的embedding。

二、跨模态对齐：



(c) Cross-Modal Alignment

1.对比学习框架

作者选择冻住CLIP的文本和图像的编码器，来保存CLIP的先验知识，防止模型坍塌。在文本和图像编码器后分别加一个线性层，和3D编码器一起训练。做对比学习时，计算3D-图像和3D-文本之间的损失。

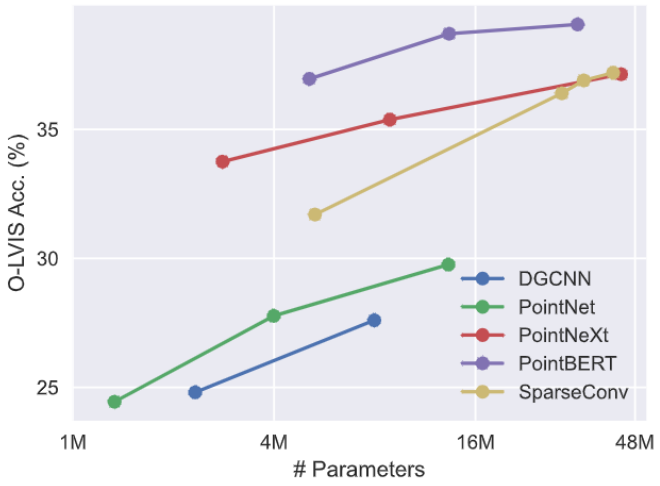
2.扩大3D骨干网络

这部分包含两个方面：扩大数据集规模和扩大模型参数。

1）通过扩大数据集规模，在shapeNet数据集上大差不大，但是在自己的数据集上可以发现不同backbones在zero-shot分类任务上表现差距很大，猜测可能是有些模型在小数据集上已经饱和了，在大数据集上无法学到更多知识。

Model	#Param.	Train on ShapeNet [8]		Train on Ens-no-LVIS	
		MNet40	O-LVIS	MNet40	O-LVIS
PointNet [51]	1.3M	67.0	9.3	74.9	24.4
DGCNN [73]	2.3M	67.8	9.0	74.2	24.8
PointMLP [42]	9.3M	73.5	12.9	82.9	36.6
PointNeXt [54]	2.8M	72.6	12.2	81.6	33.8
PointBERT [81]	5.1M	70.3	10.8	84.5	37.0
SparseConv [10]	5.3M	70.7	10.6	78.8	31.7
std. dev.		2.3	1.4	3.9	5.1

2）通过增加模型参数，可以从下图中看见模型的表现都会提升。但是扩大模型规模会导致训练时间和内存消耗快速增加，因此后面的实验只选择了基于transformer的PointBERT和基于CNN的SparseConv进行实验。



3.硬负样本挖掘

这部分主要是解决类间样本不同类别数量分布不均的问题。

因为在同一个batch中，同时出现令人困惑的样本的可能性较小。在第一轮训练中，使用随机批次进行训练，直至接近收敛。在第二轮训练中，每个批次计算每个形状嵌入的k最近邻。随机选择s个种子形状，然后获取m个邻居，每批次结果为s * m个形状。对于错误负样本（两个苹果），使用图像和文本嵌入进行过滤：如果 $h_j^T * \cdot * h_i^I * + * \delta * > * h_i^T * \cdot * h_i^I$ ，则判断i和j的文本非常接近，从i的负样本中移除j。

3.实验效果？局限性？未来展望？

本文做了三个实验：zero-shot分类任务、、消融实验

1.zero-shot分类任务

评测的数据集包括ModelNet40, ScanObjectNN, Objaverse-LVIS，如下图所示。作者发现他们的方法超过了别的方法，体现出文本处理、训练策略等的优越性；明显提升了长尾数据集上分类性能；在从模拟环境的数据集迁移到现实环境的数据集上，也有还不错的效果。

Table 2: Zero-shot classification on Objaverse-LVIS [12], ModelNet40 [75], and ScanObjectNN [70].

Method	training shape source	Objaverse-LVIS [12]			ModelNet40 [75]			ScanObjectNN [71]		
		Top1	Top3	Top5	Top1	Top3	Top5	Top1	Top3	Top5
PointCLIP [86]	2D inferences, no 3D training	1.9	4.1	5.8	19.3	28.6	34.8	10.5	20.8	30.6
PointCLIP v2 [88]		4.7	9.5	12.9	63.6	77.9	85.0	42.2	63.3	74.5
ReCon [53]	ShapeNet	1.1	2.7	3.7	61.2	73.9	78.1	42.3	62.5	75.6
CG3D [19]		5.0	9.5	11.6	48.7	60.7	66.5	42.5	57.3	60.8
CLIP2Point [24]		2.7	5.8	7.9	49.5	71.3	81.2	25.5	44.6	59.4
ULIP-PointBERT (Official) [78]		6.2	13.6	17.9	60.4	79.0	84.4	51.5	71.1	80.2
OpenShape-SparseConv		11.6	21.8	27.1	72.9	87.2	93.0	52.7	72.7	83.6
OpenShape-PointBERT		10.8	20.2	25.0	70.3	86.9	91.3	51.3	69.4	78.4
ULIP-PointBERT (Retrained)	Ensembled (no LVIS)	21.4	38.1	46.0	71.4	84.4	89.2	46.0	66.1	76.4
OpenShape-SparseConv		37.0	58.4	66.9	82.6	95.0	97.5	54.9	76.8	87.0
OpenShape-PointBERT		39.1	60.8	68.9	**85.3**	96.2	97.4	47.2	72.4	84.7
ULIP-PointBERT (Retrained)	Ensembled	26.8	44.8	52.6	75.1	88.1	93.2	51.6	72.5	82.3
OpenShape-SparseConv		43.4	64.8	72.4	83.4	95.6	97.8	**56.7**	78.9	88.6
OpenShape-PointBERT		**46.8**	**69.1**	**77.0**	84.4	**96.5**	**98.0**	52.2	**79.7**	**88.7**

2.Few-Shot Linear Probing

实验效果如下图所示。在长尾分布的LVIS上，他们的zero-shot性能超过了其他方法few-shot的性能；但是在其他数据集上，和ULIP的性能差不多。作者推测ModelNet40错误的主要来源是类内的差距大，ScanObjectNN是训练集和测试集间的domain gap大，不能说明他们没有学到很好的表征。

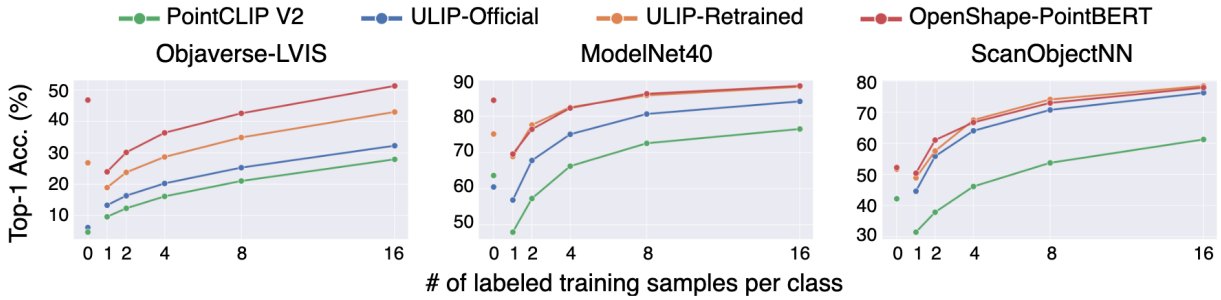


Figure 5: Few-shot linear probing on Objaverse-LVIS [12], ModelNet40 [75], and ScanObjectNN [70]. We report the average performance over 10 random seeds.

3.消融实验

从数据处理、模型规模、对比损失、训练策略等方面做了消融实验，结果如下两图所示。其中比较有意思的是，在长尾分布的LVIS数据集上，采用1%集成数据（8k形状）的效果和不用objaverse（77k形状）的性能差不多，说明在提高对未见过的形状的泛化性上，**多样性比数据规模更重要**。

Table 3: Ablation study. Top 1 zero-shot accuracies on ModelNet40 [75] and Objaverse-LVIS [12] are shown.

Variant	O-LVIS	MNet40
No Objaverse shapes	13.9	75.5
Only Objaverse shapes	41.6	79.2
No backbone scale up	31.7	78.7
No caption & retrieval	37.0	82.9
No text filtering	41.4	82.9
No point rgb, only xyz	39.6	83.6
No text contras. learning	23.3	67.4
No image contras. learning	41.0	81.0
Full	42.0	83.1
Full + hard mining	43.4	83.4

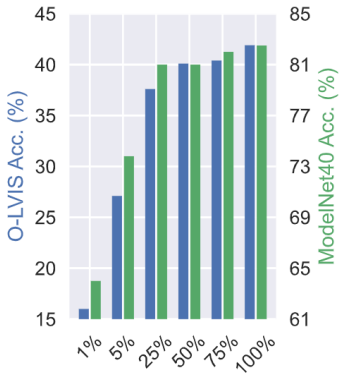


Figure 6: Ablation study on using different ratios of training data.

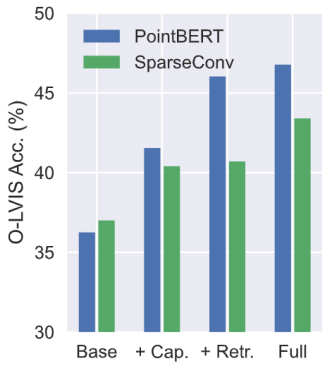


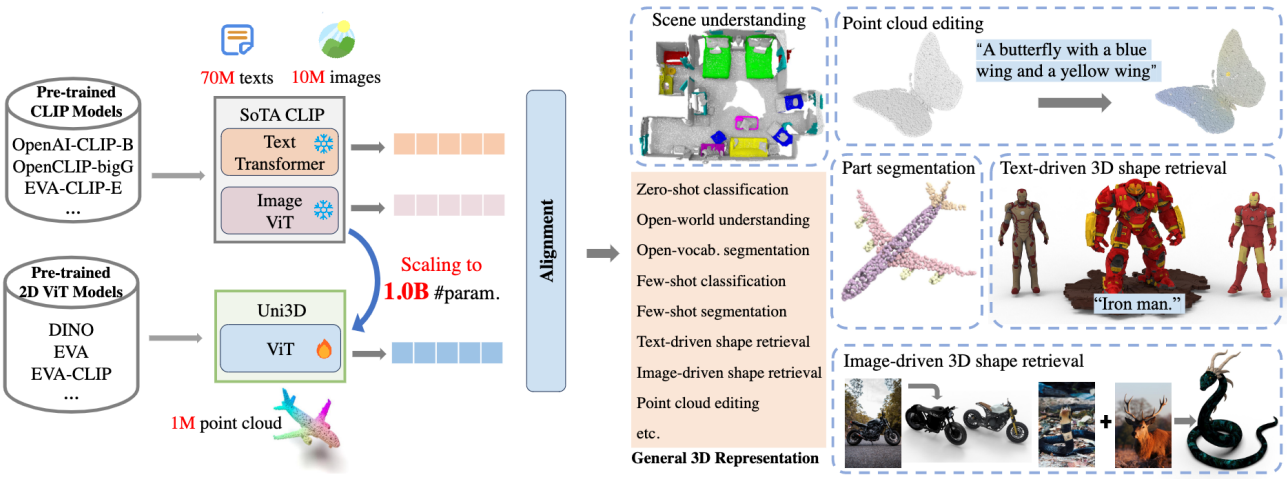
Figure 7: Ablation study on different text enrichment strategies.

ULIP2

Xue L, Yu N, Zhang S, et al. ULIP-2: Towards Scalable Multimodal Pre-training For 3D Understanding[J]. arXiv preprint arXiv:2305.08275, 2023.

ULIP-2的方法主要是：为每个3D物体生成多角度不同的语言描述，然后用这些描述来训练模型，使3D物体、2D图像、和语言描述在特征空间对齐一致。

UNI3D



Uni3D 通过以下方式处理不同模态的输入并实现图像、本和点云的对齐：

1. **统一的3D表示**：Uni3D 使用一个结构上等价于2D Vision Transformer (ViT) 的统一原始变换器作为骨干网络。它替换了ViT中的patch嵌入层，使用特定的点标记器实现3D嵌入。这使得 Uni3D 能够在3D表示学习中轻松利用2D预训练模型作为初始值。
2. **缩放Uni3D**：Uni3D 直接使用与ViT结构上等价的原始变换器作为3D骨干网络，这使得它可以简单地使用已经研究过的统一2D/NLP缩放策略进行缩放。这意味着 Uni3D 可以自然地采用在其他模态中与我们相同的原始变换器相共享的预训练大模型来初始化，例如2D预训练模型 DINO、EVA、EVA-02 以及跨模态模型 CLIP、EVA-CLIP 等。
3. **多模态对齐**：Uni3D 通过训练一个用于学习多模态对齐的3D编码器，将3D点云特征与图像-文本对齐特征对齐。这使得 Uni3D 能够在训练过程中学习到强大的3D表示，从而在各种下游任务和场景中表现出良好的性能。

Uni3D 成功地处理了不同模态的输入，并实现了图像、本和点云之间的对齐。

uni3D相当于是在Openshape的基础上做出一些优化改进和融合。