用于三维形状识别的多视角变换网络论文的复现

摘要

之前的多视图投影方法表明，在 3D 形状识别方面这些多视角投影方法能够达到最好的结果。这些方法学习从多个视图进行信息融合，但是这些视图的相机视点往往是针对所有形状进行启发式设置和固定。因此，为了避免当前多视图方法缺乏活力，本文复现的论文认为应该对这些视点进行学习，即学习如何选择这些视点。为此，该论文引入了多视角变换网络 (MVTN)，其基于可微渲染来回归出 3D 形状识别的最佳视点。MVTN 能够与任何多视图网络一起进行端到端的训练，从而进行 3D 形状分类。同时，其将 MVTN 集成到一种新的自适应多视图管道中，而该管道能够渲染 3D 网格或点云。由于 MVTN 中的点的采样较为单一，因此本文在该方法上增加了多尺度采样，以增加 MVTN 在不同点云尺度下的视点的预测，同时用自注意力点网络来替代原本的采样点的特征提取模块，从而提高网络的性能。从实验结果来看，其在分类任务上达到了一个可比较的结果。

关键词： 3D 形状识别；可学习的视点预测；端到端

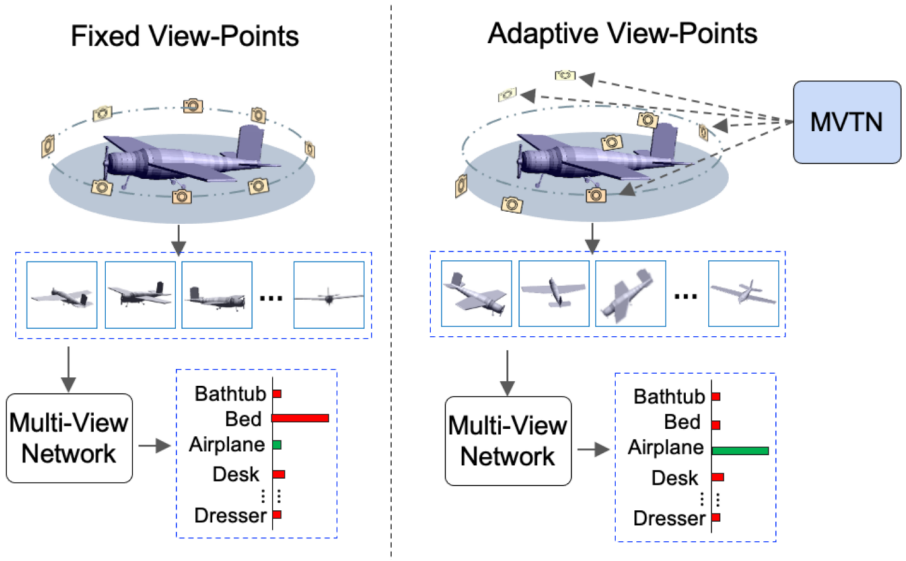

图 1. Multi-View Transformation Network (MVTN)。

1 引言

由于深度学习在 2D 领域的成功，其自然而然地扩展到了 3D 视觉领域。在 3D 中，深度网络在分类、分割和检测方面取得了很好的结果。3D 深度学习网络直接对 3D 数据进行操作，

其中的 3D 数据通常表示为点云、网格或体素。然而，其他方法选择通过渲染对象或场景的多个 2D 视图来表示 3D 信息，而这种多视图方法更类似于人类的方法，其中人类的视觉系统提供渲染图像流，而不是更复杂的 3D 表示。

多视图方法的最新发展展现出令人印象深刻的结果，并在许多情况下对于 3D 形状分类和分割方面取得了最先进的结果。多视图方法通过使用 2D 卷积架构解决 3D 任务，弥补了 2D 和 3D 学习之间的差距。这些方法为给定的 3D 形状渲染出多个视图，并利用渲染的图像来解决最终的任务。因此，他们以基于 2D 网格的深度学习的最新进展为基础，并利用更大的图像数据集进行预训练（例如 ImageNet [13]），以弥补标记 3D 数据集的普遍稀缺性。然而，此类方法如何选择渲染视点的方式大多数仍然是未被探索的。当前的方法依赖于启发式方法，如场景中的随机采样或定向数据集中的预定义规范视点，但是没有证据表明这种启发式方法在经验上是最佳的选择。

为了解决这个缺陷，本文复现论文认为通过引入多视角变换网络（MVTN [5]）来学习更好的视点。如图 1 所示，MVTN 学习如何回归视点，使用可微渲染器渲染这些视图，并以端到端的方式训练下游特定任务的网络（如 3D 形状分类），从而得到最适合该任务的视图。

2 相关工作

2.1 基于 3D 数据的深度学习

PointNet [11] 是第一个直接在 3D 点云上运行的深度学习算法，其为后面的研究奠定了基础。PointNet 单独计算点的特征，并使用最大池化这样的阶数不变函数来聚合它们，且随后的工作重点是寻找点的邻域来定义点卷积运算。基于体素的深度网络使用 3D CNN，但会受到立方内存复杂性的影响，而最近的几项工作将点云表示与其他 3D 模式（如体素 [10] 或多视图图像 [6,17]）相结合。在复现的论文中，其利用点编码器来预测最佳视点并从中渲染出图像并将其传输到多视图网络。

2.2 多视图 3D 形状分类

第一个使用 2D 图像识别 3D 物体的工作是由 Bradski 等人 [2] 提出的，而在二十年后，深度学习在 2D 视觉任务中取得成功后，MVCNN [14] 首次将深度 2D CNN 用于 3D 对象识别。原始的 MVCNN 使用最大池化来聚合来自不同视图的特征。一些后续工作提出了不同的策略来为视图分配权重，以执行视图特定特征的加权平均池化。RotationNet [7] 对视图和对象进行联合分类，等变多视角网络 [4] 通过利用旋转组卷积在多视图上使用旋转等变卷积运算，ViewGCN [15] 最近的工作利用动态图卷积运算来自适应地池化来自不同固定视图的特征，以完成 3D 形状的分类任务，这之前的方法都依赖于 3D 对象的固定渲染数据集。[3] 的工作尝试通过强化学习和 RNN 自适应地选择视图，但这取得的效果有限，而且训练过程也很复杂。在复现论文中，其提出了一种新颖的 MVTN 框架，用于预测多视图设置中的最佳视点。这是通过与多视图任务特定网络联合训练 MVTN 来完成的，不需要任何额外的监督，也不需要调整学习过程。

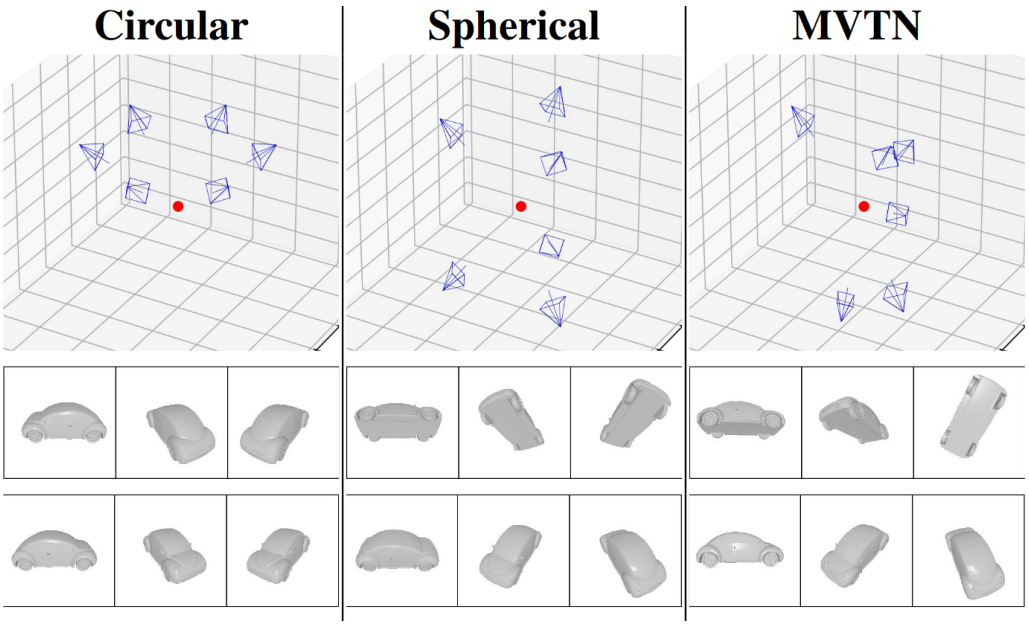

图 2. 多视角相机位置配置。

2.3 视觉作为逆向图形 VIG

VIG 的一个关键问题是经典图形管道的不可微性。最近的 VIG 方法专注于使图形操作可微，允许梯度直接从图像流到渲染参数。NMR [8] 通过平滑边缘渲染来近似不可微分光栅化，其中 SoftRas [9] 将所有网格三角形的概率分配给图像中的每个像素。Synsin [16] 提出了一种用于可微点云渲染的 alpha 混合机制。Pytorch3D [12] 渲染器提高了 SoftRas 和 Synsin 的速度和模块化性，并允许定制着色器和点云渲染。MVTN 利用可微分渲染的方法，以端到端的方式与多视图网络联合训练。使用网格和点云可微分渲染使 MVTN 能够处理 3D CAD 模型和更易于访问的 3D 点云数据。

3 本文方法

3.1 多视角网络的训练

最简单的深度多视角分类器是 MVCNN，其分类网络$C = MLP(max_i f(x_i))$, 这里的$f : \mathbb{R}^{h\times w\times c} \to \mathbb{R}^d$ 作为 2D CNN 的主干网络（即 ResNet），用来对各个渲染的图像进行分类。而若采用最近的方法 ViewGCN，则$C = MLP(cat_{GCN} f(x_i))$，这里的$cat_{GCN}$ 用于从图像卷积网络中进行视角的特征融合。一般来说，在一个有标签的 3D 数据集上，学习一个具体任务的多视角网络可被定义为：

$$\arg\min_{\theta_C} \sum_{n}^{N} L(C(X_n), y_n) = \arg\min_{\theta_C} \sum_{n}^{N} L(C(R(S_n, u_0)), y_n) \tag{1}$$

这里的 L 为具体任务的损失，y_n 是对应的 3D 形状S_n 的标签，而 $u_0 \in \mathbb{R}^\tau$ 是用于整个数据集的 τ 个固定的场景参数。这些参数 R 是一个接收形状 S_n 和 u_0 来对每个形状渲染出 M 个多视角图像 X_n 的渲染器。本文复现的论文中，场景参数 u 设置为各个相机视点对于物体中心的方向角与仰角，因此 $\tau = 2M$。

3.2 规范视角

先前的多视角方法依赖的场景参数 u_0（即相机视角参数）对于整个 3D 数据集来说是预先定义的，即一般的设置为一个圆周均匀的采样多个视点，或在一个上半球上均匀的采样多个视点，又或者是凭借经验来指定某些视点。这些预先定义的视点位置在某些情况下会造成一些负面影响，因此本文复现的方法通过学习如何回归出这些视点的位置，如图2所示。

3.3 多视角变换网络 (MVTN)

3.3.1 可微分渲染器

一个渲染器 R 通过接收 3D 形状与场景参数 u 作为输入来渲染出对应的 M 对应的图像 $\{x_i\}_{i=1}^{M}$ 。由于 R 是可微的，因此场景参数 u 的梯度可以通过渲染的图像进行反向传播得到，故能够建立起一个端到端的训练过程。对于输入形状是 3D 网格还是点云来采用不同的渲染方法，渲染的结果如图2与图3所示。

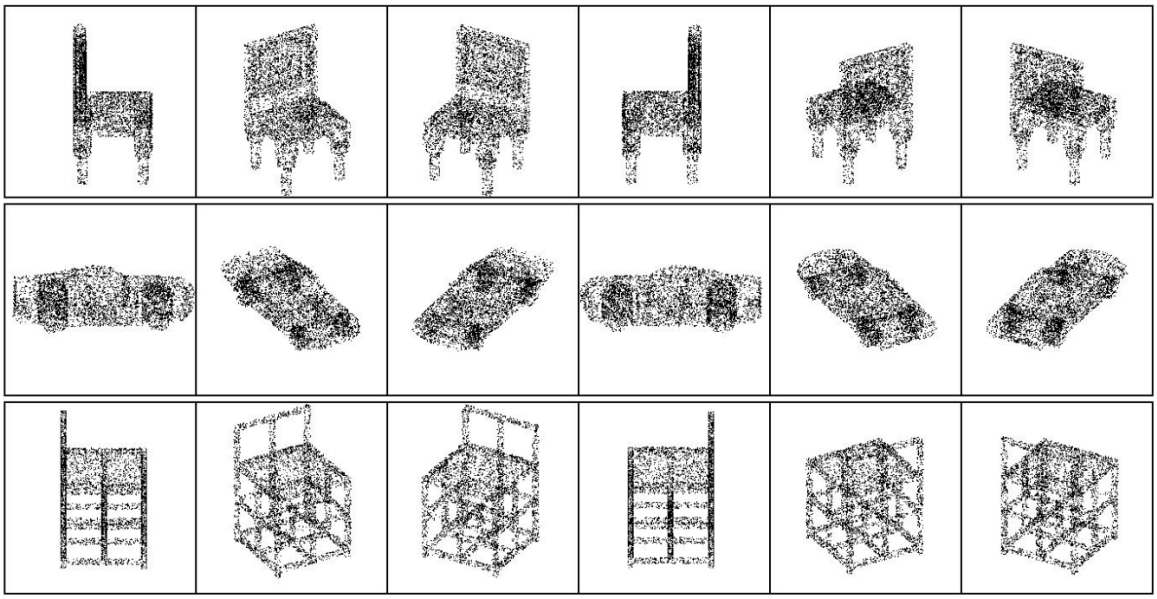

图 3. 多视角点云渲染。

3.3.2 对 3D 形状的视点限制

本文复现的论文设计 u 为一个 3D 形状函数，其通过 MVTN 来进行拟合。具体来说，通过 MVTN 在接收一些从形状中采样的采样点 P 来回归出 u 的具体值。端到端的训练过程通过最小化以下损失函数：

$$\arg\min_{\theta_C,\theta_G} \sum_{n}^{N} L(C(R(S_n,u_n)),y_n),\ s.t.\ u_n = u_{bound}\cdot tanh(G(S_n)) \tag{2}$$

这里的 G 网络通过对从 3D 形状中得到的采样点来预测出能够更好的服务于下游的多视角分类网络 C 的最优视点。由于网络 G 的目标只预测视点而不进行分类，因此可以使用简单的点编码器（例如在 PointNet 中的一个共享的 MLP 网络）来对采样点进行特征提取即可。然

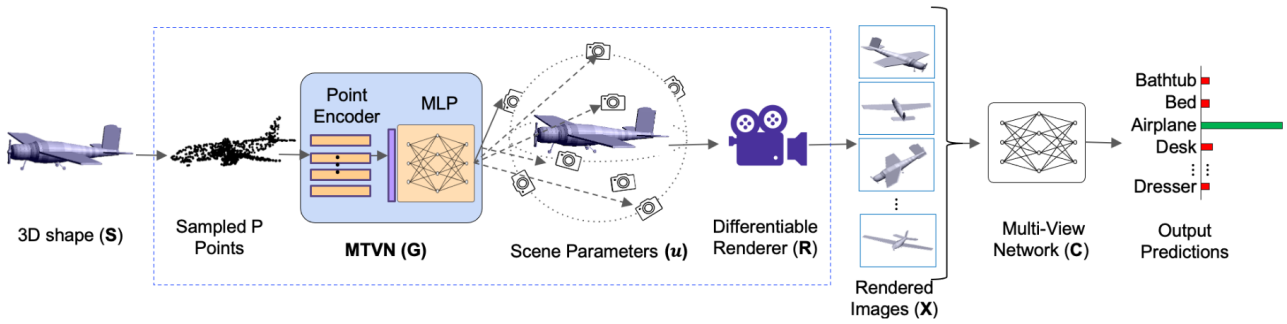

图 4. 用于多视角分类的端到端训练。

后，通过一个轻量的 MLP 网络来回归出场景参数 u_n，即方向角与仰角。由于方向角与仰角需要再一个合理的范围内，因此这里需要对预测的结果加以约束，这里通过使用双曲正切函数以及缩放因子 u_{bound} ，使 u_n 范围在 $\pm u_{bound}$。

3.3.3 MVTN 用于 3D 形状分类

为了训练用于 3D 形状分类的 MVTN，本文复现论文在公式 (2) 中定义了一个交叉熵损失，同样也能够在这里采用其他的损失和正则化项。由于端到端的训练，多视角网络（C）与 MVTN（G）在同一个损失函数上进行训练，同时端到端的处理 3D 点云也是本文复现论文的一大亮点，如图4所示。

4 复现细节

4.1 与已有开源代码对比

本文在 MVTN 源码中的单一尺度采样的基础上增加为多尺度采样，同时采用了与 Point tnt [1] 中类似的点 Transformer，但通过减少 Transformer 序列的数量来使网络变的更加轻量化，以此来保证在算力资源有限的情况下能够正常的进行训练。

4.2 实验环境搭建

4.2.1 运行环境

操作系统	Ubuntu 20.04
CPU	Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz 220GB　程序运行与数据处理
GPU	NVIDIA TITAN Xp 12GB, RTX 3090 24GB　模型的训练加速
编程语言	Python 3.7.16
DL 开发平台	Pytorch 1.7.0+cu110

4.2.2 配置环境

- 安装 Anaconda，创建 Python 3.7.16 的虚拟环境

- 安装运行 MVTN 的所有 Python 包

4.3 界面分析与使用说明

4.3.1 模型训练

点编码器的修改以及单尺度采样与多次度采样的方法需要在源码中进行修改。

```
python run_mvtn.py --data_dir data/ModelNet40/ --run_mode train --mvnetwork
                                                mvcnn --nb_views 6 --views_config
                                                learned_spherical
```

4.3.2 模型测试

测试模型时需保证已经训练过模型。

```
python run_mvtn.py --data_dir data/ModelNet40/ --run_mode test_cls --
                                                mvnetwork mvcnn --nb_views 6 --
                                                views_config learned_spherical
```

4.4 创新点

由于原文的采样方式比较单一，因此本文认为应该能够从多个尺度来对 3D 形状进行采样，以此来采集各个尺度下的点集，从而使 MVTN 在不同的尺度下预测不同的视角，从而提高总的性能；同时，由于近年来自注意力机制的发展，因此想用它来作为采样点的编码器，并采用更加轻量的 transformer 来进行实验。

5 实验结果分析

由图4可知，模型能够大致分为采样模块，MVTN 网络，渲染网络以及分类网络，其中本文对采样模块与 MVTN 模块进行的修改，其余模块基本保持不变。渲染网络采用 Pytorch 3D 中提供的可微分渲染器，而分类网络采用主干为 ResNet18 的 MVCNN。所有的模型都是训练了 100 个 Epoch，显示的都是 100 个 Epoch 中最优的分类结果。

5.1 数据集

本文在 ModelNet40 数据集上进行模型的训练以及测试。该数据集分为 40 个类别，共有 12311 个 3D 模型组成，其中 9843 个模型用作训练集，2468 个模型用作测试集。同时又由于硬件的限制，通过使用 Blender API 将这些模型被简化到两万个顶点。

5.2 对比原文的结果

方法	视角数	批训练数	模型参数量	点编码器	准确率
MVTN	6	20	3.31M	PointNet	91.94%
MVTN	6	20	0.91M	DGCNN	91.29%
MVTN	12	20	3.32M	PointNet	91.82%
MVTN	12	20	0.91M	DGCNN	91.98%
Ours（实验 3）	6	4	1.71M	Transformer	91.96%
Ours（实验 4）	12	4	1.71M	Transformer	91.50%

表 1. 采用多尺度采样与 Transformer 点编码器的结果。

通过表1可知，本文提出的方法能够在视角数为 6 个的情况下达到较好的效果，而在视角数为 12 个的情况下，却也只有不到 0.5% 的差距。可以注意到，本文模型训练时所采用的批训练数较小，这是由于资源的限制问题，或许在批训练数更高的情况下能够达到更好的效果。同时，由于渲染网络与分类网络一致，因此这里的模型参数量只考虑了 MVTN 视角选择这一模块，并相较于 PointNet 点编码器，我们参数量也少了进一半。

5.3 消融实验

方法	视角数	批训练数	模型参数量	点编码器	准确率	备注
实验 1	3	5	3.59M	Transformer	90.11%	序列数：4 头数：3
实验 2	3	5	1.75M	Transformer	89.84%	序列数：1 头数：3
实验 3	6	4	1.71M	Transformer	91.96%	序列数：1 头数：2
实验 4	12	4	1.71M	Transformer	91.50%	序列数：1 头数：2
实验 5 (No multi-sample)	3	5	1.71M	Transformer	88.57%	序列数：1 头数：2
实验 6 (No multi-sample)	3	5	1.75M	Transformer	89.95%	序列数：1 头数：3
实验 7	3	5	3.31M	PointNet	90.46%	None
实验 8 (No multi-sample)	3	5	3.31M	PointNet	88.41%	None

表 2. 不同编码器以及是否进行多尺度采样的结果。

实验结果如表2所示。由实验 1 与实验 2 可知，不同的 Transformer 序列数对于准确率有着一定的影响；由实验 7 与实验 8 可知，多尺度采样嫩能够对结果产生一定的提高，而采样

的具体策略为将原来只采 2048 个点转为分别采 1024, 2048 以及 4096 个点；通过实验 5 与实验 8 能够反映出采用 Transformer 点编码器是具有效果的；而通过实验 5 与实验 6 可以看出 Transformer 头的数量会对最终的结果产生一定的影响。总的来说，在资源有限的情况下设计一个轻量又不失性能的模型至关重要。

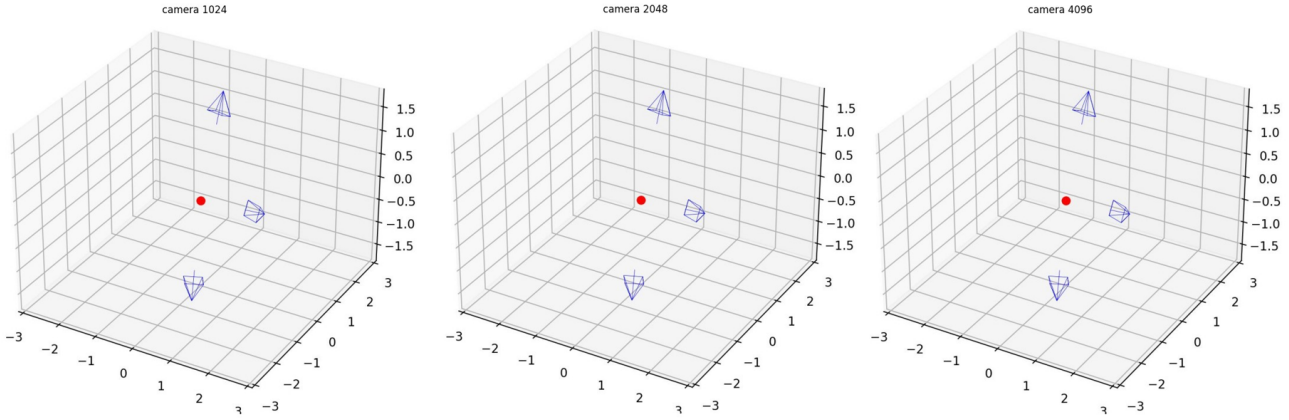

图 5. 不同尺度下的视角选择结果。

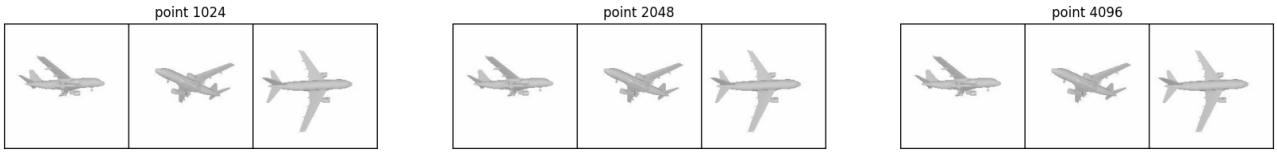

图 6. 不同尺度下的视角选择后的渲染结果。

视角的选择如图5与图6所示，可以发现在不同尺度下的采样点进行视角的选择几乎没有什么变化，而通过上述表2中的实验 7 与实验 8 表明使用多尺度采样确实能够提升一定的效率。

6 总结与展望

本文中提出的多尺度采样方法具有一定的随机性，因为采多少次样本以及每次采多少个点，这些都是无法确定的，因此都是事先定义好的。以此同时，是否使用 Transformer 作为点编码器也具有一定的局限性，因为在资源有限的情况下只能对其进行轻量化处理，所有很有可能无法发挥原模型具有的特性，并且是否只有在某种特定的条件下采用 Transformer 作为点编码器才能发挥更好的作用还有待考证。综上所述，我认为在原来的训练过程中启发式点采样会被有可能会被自适应点采样所替代，因为原来的启发式点采样缺少合理性，同时也无法保证最优性，故这是在未来可能是一个研究方向。

参考文献

[1] Axel Berg, Magnus Oskarsson, and Mark O' Connor. Points to patches: Enabling the use of self-attention for 3d shape recognition. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 528–534. IEEE, 2022.

[2] Gary Bradski and Stephen Grossberg. Recognition of 3-d objects from multiple 2-d views by a self-organizing neural architecture. In *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, pages 349–375. Springer, 1994.

[3] Songle Chen, Lintao Zheng, Yan Zhang, Zhixin Sun, and Kai Xu. Veram: View-enhanced recurrent attention model for 3d shape classification. *IEEE transactions on visualization and computer graphics*, 25(12):3244–3257, 2018.

[4] Carlos Esteves, Yinshuang Xu, Christine Allen-Blanchette, and Kostas Daniilidis. Equivariant multi-view networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1568–1577, 2019.

[5] Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. Mvtn: Multi-view transformation network for 3d shape recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2021.

[6] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointnet for 3d scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[7] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5010–5019, 2018.

[8] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916, 2018.

[9] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019.

[10] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[11] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[12] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020.

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[14] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[15] Xin Wei, Ruixuan Yu, and Jian Sun. View-gcn: View-based graph convolutional network for 3d shape analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1850–1859, 2020.

[16] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020.

[17] Haoxuan You, Yifan Feng, Rongrong Ji, and Yue Gao. Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1310–1318, 2018.