

BrepGen

论文：《BrepGen: A B-rep Generative Diffusion Model with Structured Latent Geometry》

地址：<https://arxiv.org/abs/2401.15563>

年份：SIGGRAPH 2024

Introduction

任务：CAD 模型生成

技术贡献：

(1) 使用树形结构来编码 CAD 模型，利用扩散模型对树节点去噪以生成 CAD 模型

Method

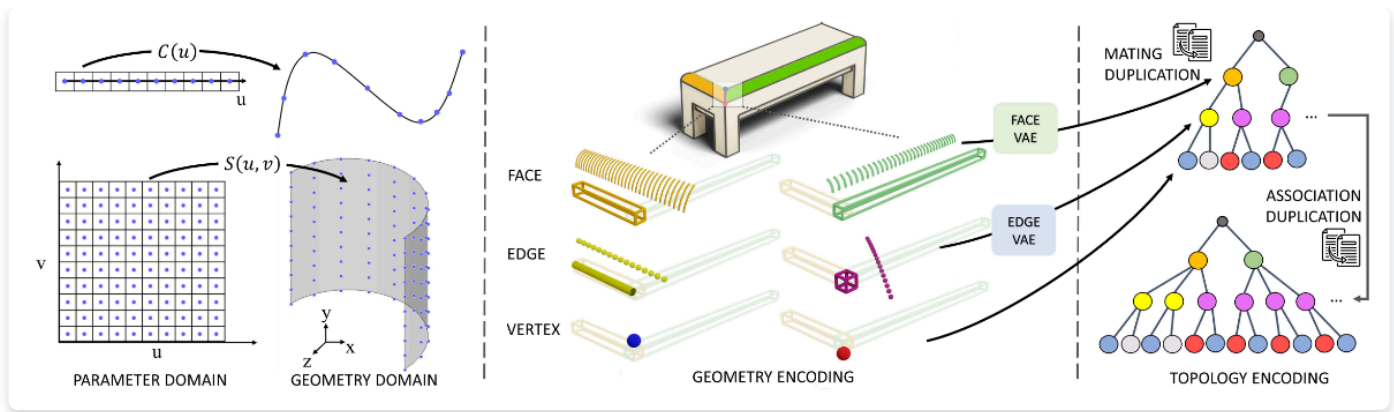
Structured Latent Geometry

使用 B-rep 表示的 CAD 模型包含三个层级：面、边和点，因此适合用树形结构来表示模型，例如面节点的子节点为边节点，边节点的子节点为顶点节点。在一个节点中，需要包含的信息有全局的位置信息和局部的形状信息。下面分别介绍三种节点中包含的信息。

- 面：每个面都是参数化曲面 $S(u, v) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ ，将 UV 坐标映射为 3D 坐标。我们将在曲面上采样的一系列点作为这个面的形状特征 F_s ，具体做法参照 UV-Net，记 $[u_{\min}, u_{\max}] \times [v_{\min}, v_{\max}]$ 为 UV 轴的 2D 包围盒，将围成的区域划分为 $N \times N$ 的网格，令网格步长为 $\delta_u = \frac{u_{\max} - u_{\min}}{N}$, $\delta_v = \frac{v_{\max} - v_{\min}}{N}$ ，采样得到的点作为曲面的特征 $F_s \in \mathbb{R}^{N \times N \times 3}$ 。同时采用 VAE，将 F_s 压缩为 F_z 。对于位置特征，就是包围这些点的包围盒的左下角和右上角坐标 $F_p = [x_1, y_1, z_1, x_2, y_2, z_2]$ ，节点的特征由这两者组成 $F = [F_p, F_z]$ 。
- 边：与面同理，沿 U 轴采样一系列点作为形状特征 E_s ，包围盒坐标作为位置参数 E_p ，也采用 VAE 压缩。 $E = [E_p, E_z]$
- 顶点：只需要考虑顶点坐标 $V = (x, y, z)$

节点中包含的特征信息编码了模型的几何特征，拓扑特征则由节点的连接关系编码。通过 2 种节点复制策略使得 B-rep 的拓扑信息变为树形结构。

- Mating Duplication：通常两个面会有共同的一条边，两条边会有共同的一个顶点，因此这些共享的元素需要进行复制，使得其所属的父节点都有一份关于其的信息，具体就通过复制来实现。
- Association Duplication：由于每个模型面的数量以及每个面的边数量通常不是固定的，变化的节点数量不利于建模。因此设置了一个最大的节点数量，小于这个数量时则选择一些节点进行复制填充，以达到最大数量。通过删除重复的节点即可还原。



Shape Geometry VAE

根据之前描述的，会使用 2 个不同的 VAE 将 F_s, E_s 压缩为 F_z, E_z ，分别通过 2D 卷积和 1D 卷积实现。设置 $N = 32$ ，下采样系数为 8，因此最终 $F_z \in \mathbb{R}^{4 \times 4 \times 3}, E_z \in \mathbb{R}^{4 \times 3}$ ，为了减少计算量，边的 2 个顶点也会与 E_z 拼接到一起，因此 $E_{zv} \in \mathbb{R}^{4 \times 3 + 6}$ 。

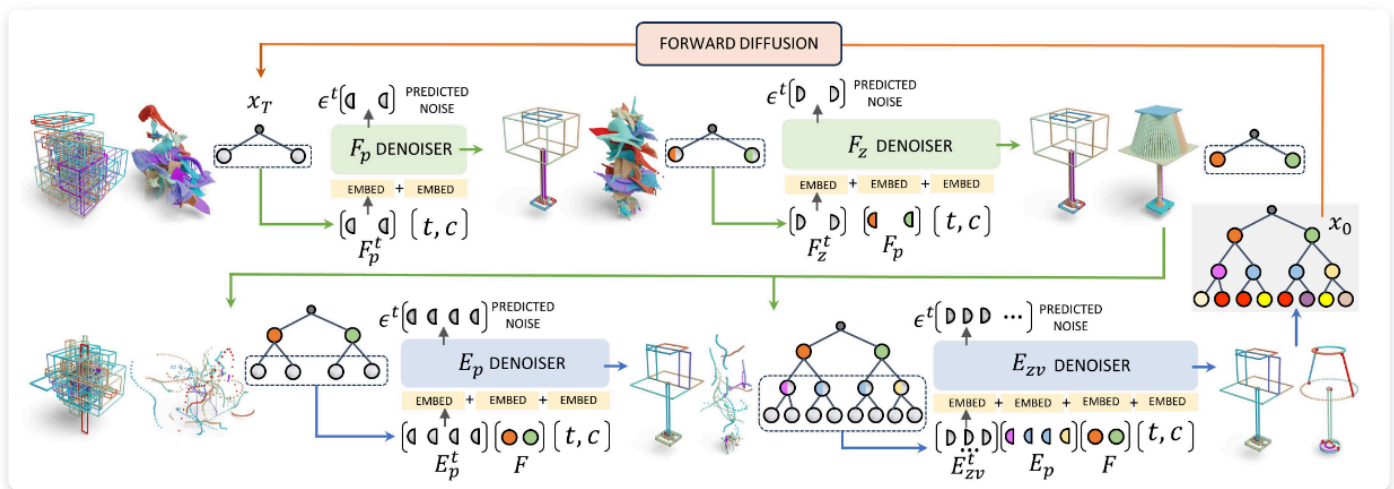
Latent Diffusion Module

Diffusion Process

参照 DDPM，对树节点特征 x_0 加噪共 T 步。

Sequential Denoising

采用自顶向下的方式进行去噪，先对面进行去噪，然后将生成的面作为条件对边进行去噪，最后将生成的面和边作为条件对顶点进行去噪。



去噪过程如上图所示，先对位置特征去噪，然后会将位置特征和面的特征 (以边的去噪为例) 作为信息注入，这里信息的注入不使用 cross-attention，而是采用直接相加的方式。

B-rep Post-Processing

这里的后处理只讲了如何对树中的节点做去重，没有将如何将树结构恢复成 B-rep 的模型。节点重复的检测取决于包围盒和采样点的相似性，只要小于一定阈值则认为是重复的节点。

Experiments

数据集: DeepCAD, Furniture B-rep

评估指标: 基于分布的 COV, MMD, JSD; 基于 CAD 的 Novel, Unique, Valid