Final Project: 3D Latent Diffusion Generation

姓名: 方嘉聪 学号: 2200017849

1 Introduction

1.1 整体介绍

参考 SDFusion [3], 先预训练 VQ-VAE, 而后是基于 Diffusion Generation 实现了 latent space 下的无条件生成与文本条件生成, 具体见下:

- VQ-VAE 使用 ShapeNet [1] 中 `chair`, `rifle`, `table`, `sofa` 和 `speaker` 5 类物体进行训练, 重建结果 **Figure** 2, 可以基本重建出物体的形状.
- 基于 VQ-VAE, 在前 4 类物体数据上分别训练了 U-Net 用于无条件生成, 结果见 Figures 3 and 5.
- 对于文本条件生成, 使用 Text2Shape [2] 数据集和预训练的文本编码器 (T5-Base [4]), 支持文本控制 `chair` 和 `table` 的生成, 结果见 Figures 4 and 7.

1.2 项目结构

整个项目结构如下:

- `data/` 与 `dataset_info_files/`: 数据集与数据集的元数据文件, 由于数据集较大, 可以参照 `README.md` 进行下载和预处理 (需要一定的时间).
- `vqvae/`: VQ-VAE 模型的实现.
- `diffusion_unet`: U-Net 模型的实现.
- `test_*.py` 与 `train_*.py`: 训练与测试脚本, 分别对应 VQ-VAE 和 U-Net 的训练与测试.
- `results/`: 生成结果的保存目录, 以 `.obj` 保存, 报告中的图片均来自该目录下的结果, 使用 MeshLab 可视化并截图.
- `log_weights/`: 训练过程中保存的模型权重及日志文件, 以便后续分析与复现. 由于模型较大, 可以从 北大网盘 下载, 解压后放置在根目录即可.
- `doc/`: 文档目录, 包含本报告的 LaTeX 源文件和结果截图 (`doc/imgs/`).

2 Methods

整体的框架见 **Figure** 1, 与 SDFusion [3] 的训练与测试流程类似, 模型具体架构与实现上由于算力和时间限制进行了调整, 具体见下:

2.1 VQ-VAE

鉴于 VQ-VAE [5] 是经典的生成模型, 本次项目没有重新实现, 在网络架构上与 SDFusion 相同, 在 `chair`, `rifle`, `table`, `sofa` 和 `speaker` 5 类物体进行训练[1]. 记编码器和解码器分别为 E_ϕ, D_τ, 输入的 T-SDF $\mathbf{X} \in \mathbb{R}^{64\times64\times64}$ (T-SDF 阈值为 0.2), 那么

$$\mathbf{z} = E_\phi(\mathbf{X}), \quad \hat{\mathbf{X}} = D_\tau(\mathrm{VQ}(\mathbf{z})) \tag{1}$$

[1] 没有使用 `car` 类别是因为数据文件大小过大, 上传到机器上需要较长时间.

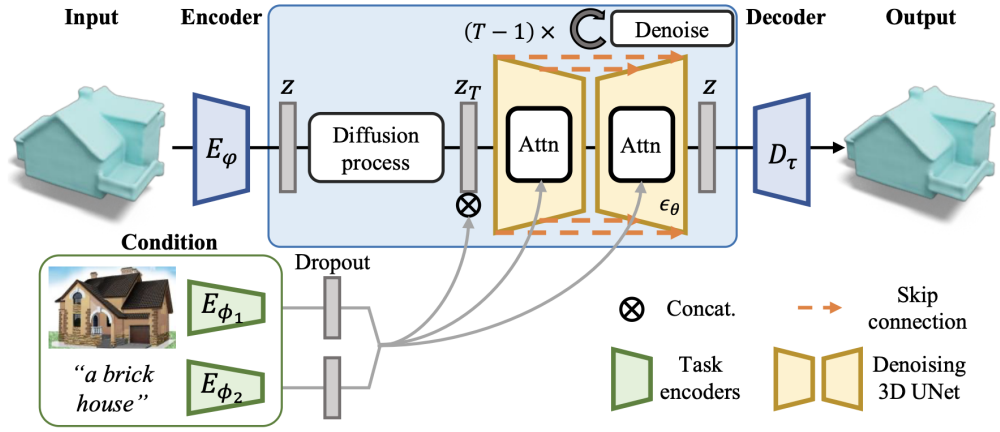


Figure 1: The overall pipeline of the project. The VQ-VAE is trained first, then the diffusion model is trained for unconditional and conditional 3D generation. The pipeline image is from SDFusion [3].

在实际训练中令 $\mathbf{z} \in \mathbb{R}^{16\times16\times16}$. 损失函数与 VQ-VAE 相同, 即

$$\mathcal{L}_{\text{VQ-VAE}} = -\log p(\mathbf{X}|\mathbf{z}) + \|\text{sg}[\text{VQ}(\mathbf{z})] - \mathbf{z}\|^2 + \|\text{VQ}(\mathbf{z}) - \text{sg}[\mathbf{z}]\|^2 \tag{2}$$

其中 sg[·] 表示停止梯度传播 (stop gradient).

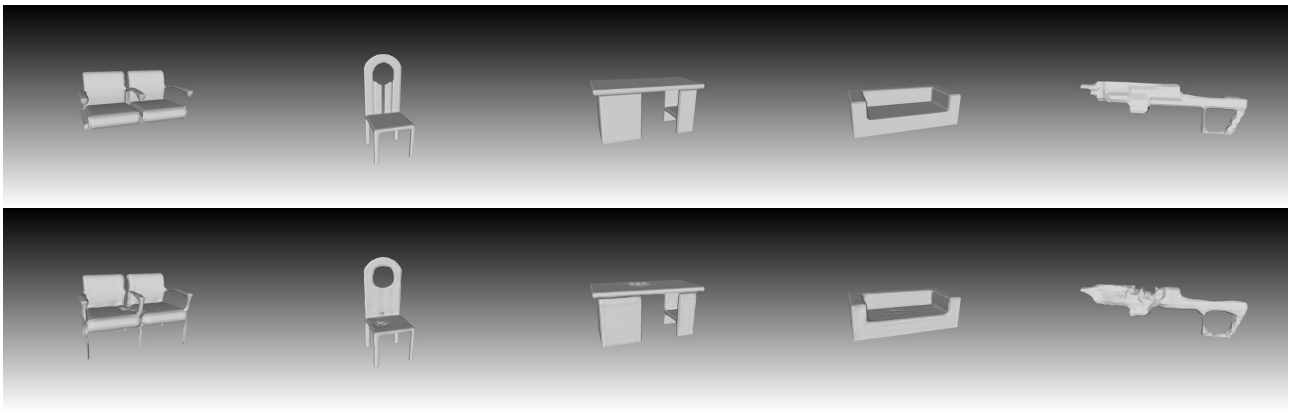


Figure 2: The reconstruction results of VQ-VAE on the test set of ShapeNet. The first row is the original object, the second row is the reconstructed object.

在训练完成后从测试集中随机采样 SDF 作为输入, 经过 VQ-VAE 重建之后的结果见 **Figure** 2. VQ-VAE 能够重建出大多数细节, 但是对于一些细节会有损失, 特别是对于 `rifle` 这类相对较小的物体. 此外还注意到生成的 `table` 会有一个凹陷处 (第三列), 没有弄清产生的原因. 在后续生成的结果中同样有部分结果有这一凹陷, 猜测还是由于 VQ-VAE 重建的信息损失导致的.

2.2 Unconditional Generation

模型架构. 对于无条件的生成, 在 [6] 的基础上进行修改, 基于 `ResidualUNet3D` 模块 (未添加 Attention Module) 实现了 UNet-Diffusion Model, 添加了必要的 time embedding 模块, 采样 scheduler 使用最简单的 DDPM. 最终使用的模型配置为:

```
"medium": {
        "time_emb_dim": 256,
        "f_maps": 128,
        "num_levels": 3,
    },
```

更细节的实现可以参考 `./diffusion_unet/model.py` 与 `./train_unet.py`, 可以通过训练脚本的命令行参数修改模型的大小.

优化目标. 在训练时, 给定任意一个 input latent $\mathbf{z}$, $\mathbf{z}_t$ 为向 $\mathbf{z}$ 添加 t 步高斯噪声 $\epsilon \in \mathcal{N}(0,1)$ 后的结果 U-Net ϵ_ϕ 的优化目标为

$$\mathcal{L}_{\text{simple}} := \mathbb{E}_{\mathbf{z},\epsilon\in\mathcal{N}(0,1)}\left[\|\epsilon - \epsilon_\phi(\mathbf{z_t}, t)\|^2\right] \tag{3}$$

测试. 随机从高斯噪声中采样 $\mathbf{z}' \in \mathbb{R}^{16\times16\times16}$, 经过 ϵ 去噪与 D_τ 解码后即可得到最终的 SDF.

2.3 Text Conditioned Generation

模型架构. 参考了 SDFusion [3] 中的一些基础模块, 相较于无条件生成, 向 U-Net 中添加了 Self-Attention 和 Cross-Attention 模块. 分为 Downsample Block, Middle Block 和 Upsample Block 三个部分, 其中 Downsample Block 和 Upsample Block 中每一次降采样 `num_res_blocks` 设置为 2, 在制定的层数上添加了 Cross-Attention 模块, 具体见 `./diffusion_unet/model_attention.py`.
文本编码器选择了 Google 的 T5-Base [4], 通过 Hugging Face `transformers` 库下载与调用. 输入的控制文本通过 T5 编码器编码为 text embedding 后, 通过 Cross-Attention 模块添加进 U-Net 中, 控制扩散过程. 注意这里与 SDFusion 不同, 没有将 text embedding 拼接到 input latent 上[2].

优化目标. 在训练时, 给定任意一个 input latent $\mathbf{z}$ 和对应的描述文本 c, 设文本编码器为 T_θ, $\mathbf{z}_t$ 为向 $\mathbf{z}$ 添加 t 步高斯噪声 $\epsilon \in \mathcal{N}(0,1)$ 的结果, 那么 U-Net ϵ_ϕ 的优化目标为

$$\mathcal{L}_{\text{text}} := \mathbb{E}_{\mathbf{z},\epsilon\in\mathcal{N}(0,1)}\left[\|\epsilon - \epsilon_\phi(\mathbf{z_t}, t, T_\theta(c))\|^2\right] \tag{4}$$

为了使用 classifier-free guidance, 在训练时以 $p = 0.1$ 的概率将 $c := \varnothing$, 即不使用文本条件.

测试. 使用 classifier-free guidance, 给定文本 c, 随机从高斯噪声中采样 $\mathbf{z}' \in \mathbb{R}^{16\times16\times16}$, 基于 c 与 $\varnothing$ 计算 $\epsilon_\phi(\mathbf{z_t}, t, T_\theta(c))$ 和 $\epsilon_\phi(\mathbf{z_t}, t, T_\theta(\varnothing))$, 最终的去噪结果为:

$$\hat{\epsilon} = \epsilon_\phi(\mathbf{z_t}, t, T_\theta(\varnothing)) + w \cdot [\epsilon_\phi(\mathbf{z_t}, t, T_\theta(c)) - \epsilon_\phi(\mathbf{z_t}, t, T_\theta(\varnothing))] \tag{5}$$

其中 w 是 classifier-free guidance 的权重, 实际测试中取 $w = 7.5$.

3 Experiment Details

所有的实验均在单卡 NVIDIA RTX 4090D 上进行, learning rate scheduler 使用 `CosineAnnealingLR`, 其中 `lr` $= 10^{-4}$, `eta_min` $= 10^{-6}$. 优化器均为 Adam, 其他超参数设置见表 **Table** 1.

[2]在实现早期遗漏了 SDFusion 的这一点, 测试结果发现已经达到预期, 没有再修改重新训练.

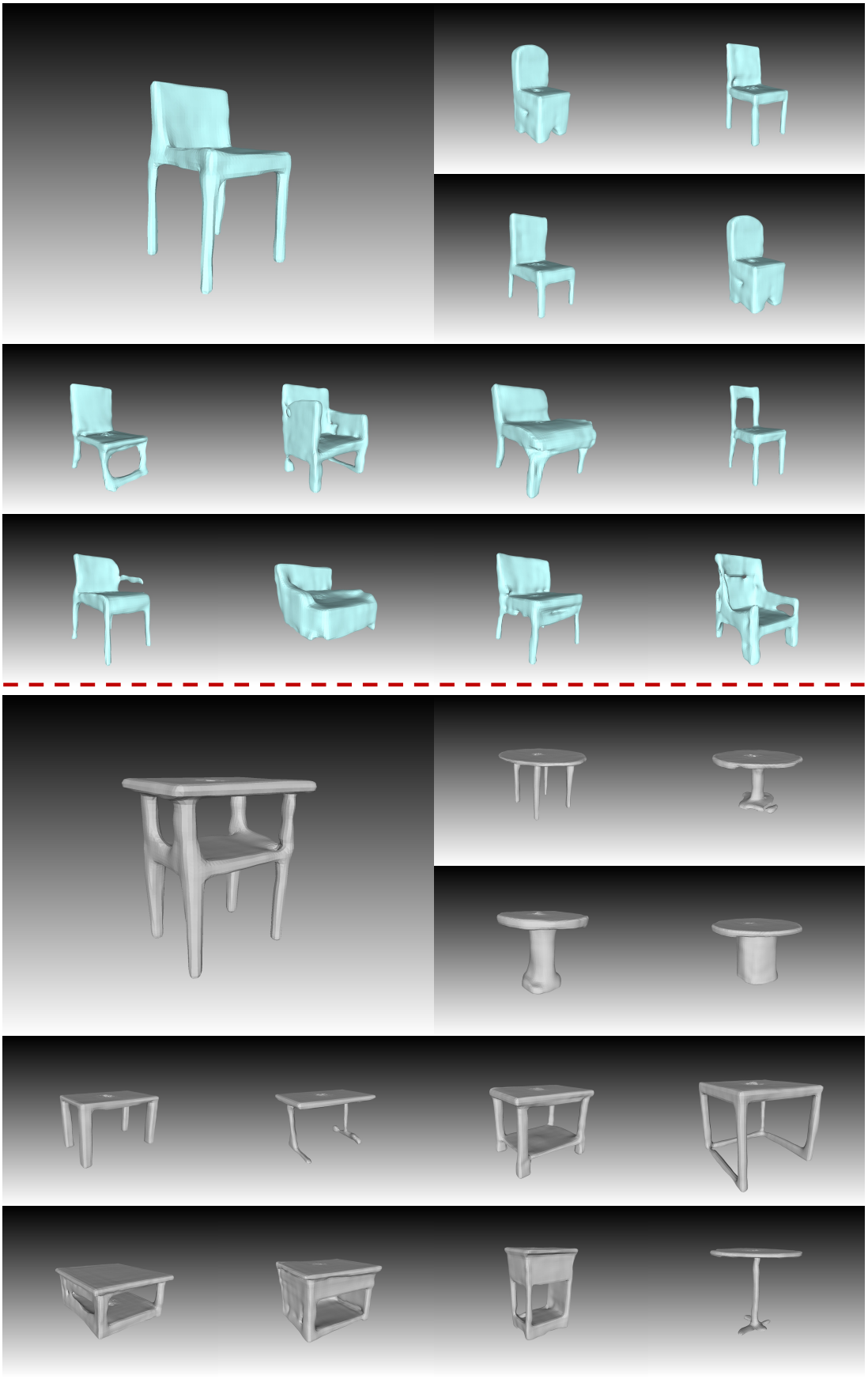


Figure 3: Unconditional 3D generation results of table and chair.

Experiments	category	batch size	#epoch	train timesteps
VQ-VAE	`all`	8	50	N/A
Unconditional	`table`	12	200	5000
	`chair`	12	200	5000
	`sofa`	12	200	5000
	`rifle`	12	400	5000
Conditional	`table, chair`	40	30	3000

Table 1: The parameters of the experiments.

Datasets. VQ-VAE 的训练数据集为 ShapeNet [1] 中的 `chair`, `rifle`, `table`, `sofa` 和 `speaker` 5 类物体, 按照 SDFusion 中的方法划分为训练集与测试集, 详见 `./dataset_info_files/`. Unconditional Generation 的针对每个类别单独训练, 得到的模型可以生成对应类别的物体, 实际上也基于 `speaker` 的数据进行了一次训练, 但是由于模型精度问题, 生成的结果基本为一个长方体, 故未在报告中展示. Text Conditioned Generation 的文本描述来自 Text2Shape [2] 数据集, 只使用了 `chair` 和 `table` 两类物体的描述, 按照 SDFusion 中的方法划分为训练集与测试集.

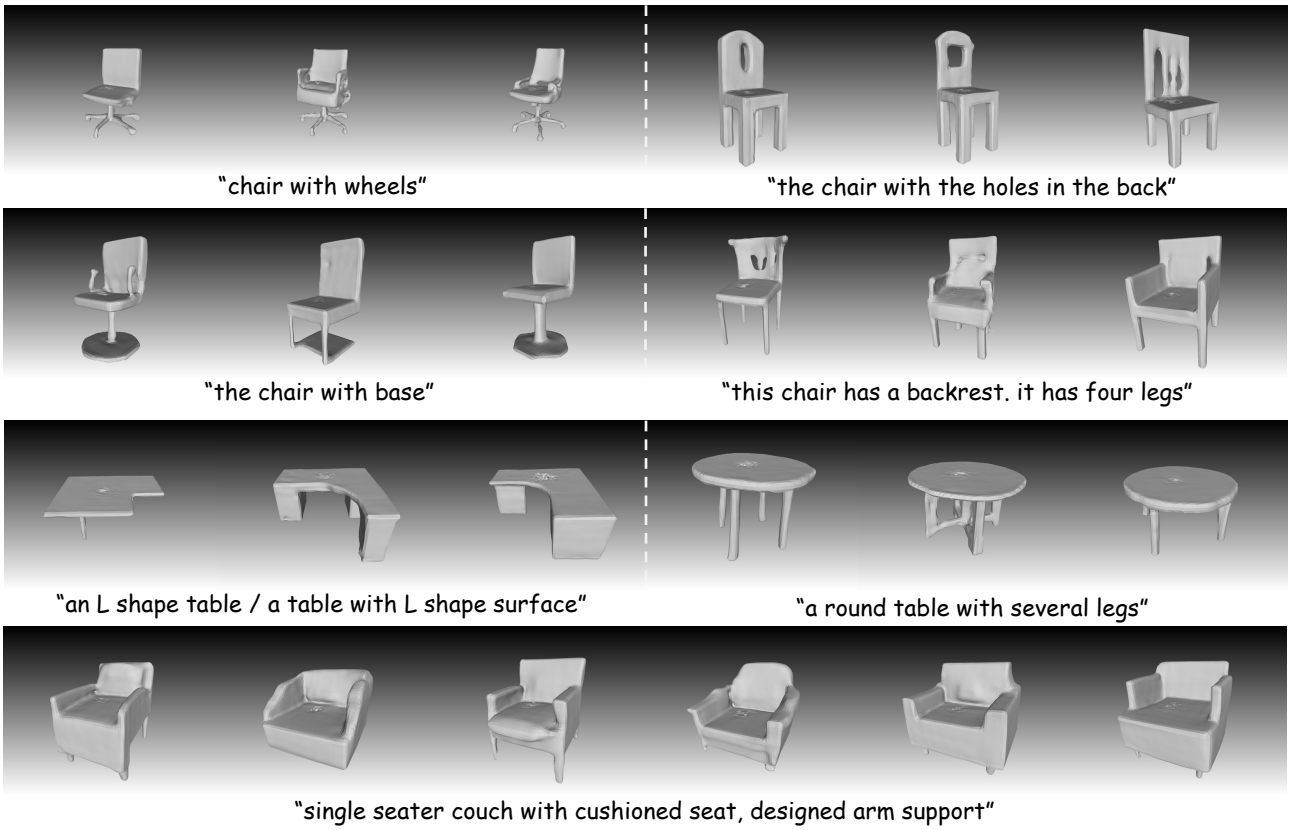


Figure 4: Conditional 3D generation results of table and chair with text prompts.

测试. 对于 VQ-VAE, 随机从测试集中采样若干 SDF, 得到重建后的结果 (**Figure** 2). 对于 Unconditional Generation, 每次随机采样若干个噪声 latent. 对于文本条件生成, 使用了 SDFusion 论文与

supplementary 中的文本描述与自己基于测试集修改的文本描述, 生成的结果见 Figures 4 and 7. 在实际测试中, Marching Cubes 的 SDF Threshold 设置在 $[0.001, 0.008]$ 之间, 去噪步数设置在 $[1000, 5000]$ 之间 (注意不能超过模型训练时的最大步数), guidance scale 设置在 $[5, 7.5]$ 之间. Noise scheduler 使用了 `diffusers` 库提供的 DDPM 接口, 可以比较方便地替换为其他 scheduler.

4 Results Analysis

4.1 Unconditional Generation

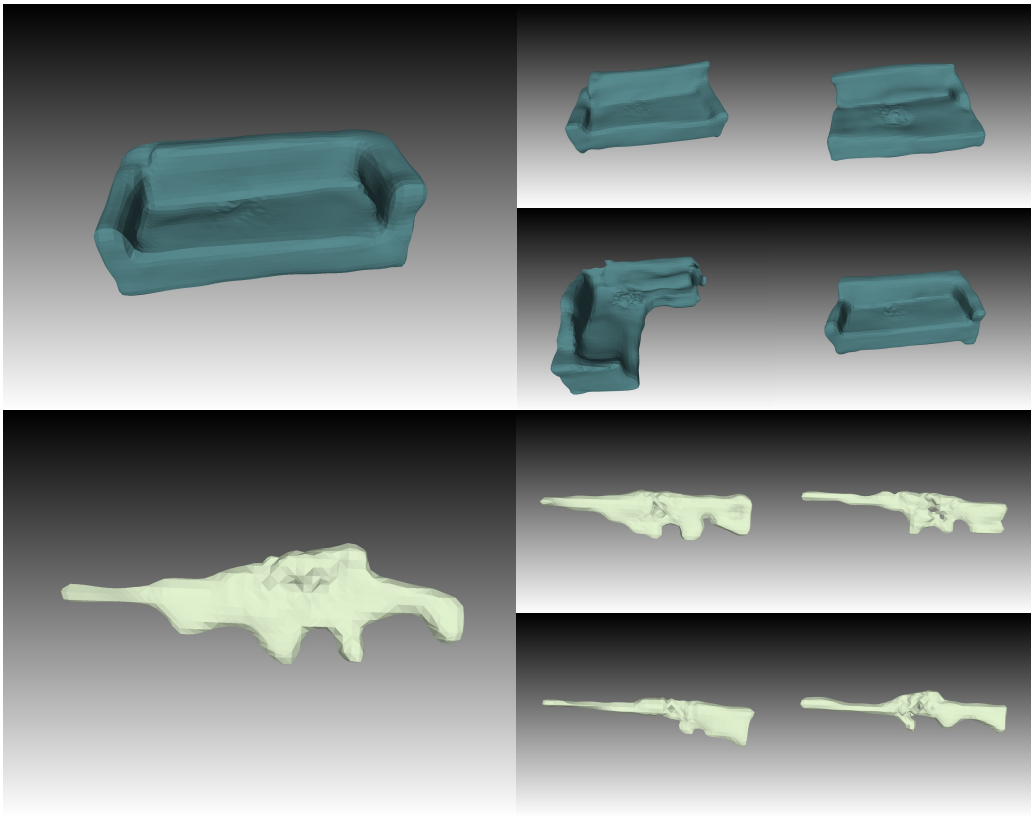


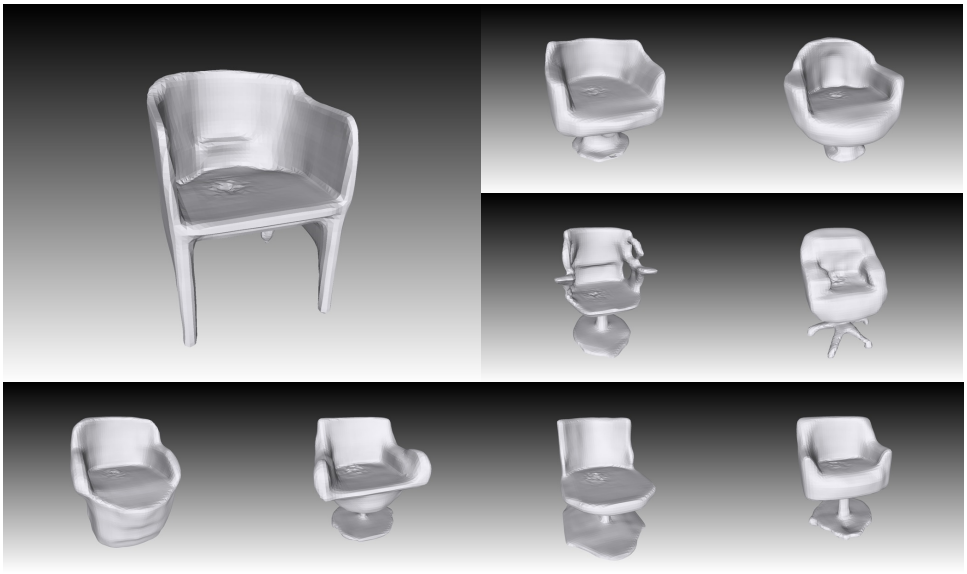
Figure 5: More unconditional 3D generation results of sofa and rifle.

Unconditional Generation 的结果见 Figures 3 and 5, 发现可以生成比较多样的物体形状, 如可以生成各类不同形状的椅子与桌子, 同时几何形状也比较合理. 但细节的生成效果不佳, 例如 `chair` 的腿部细节 (见 **Figure** 6) 及 `rifle` 的整体生成细节 (见 **Figure** 5) 较差. 一部分可能原因是 VQ-VAE 重建的 SDF 信息损失. 另一部分可能是由于 `rifle` 物体较小且相对数据量较小.

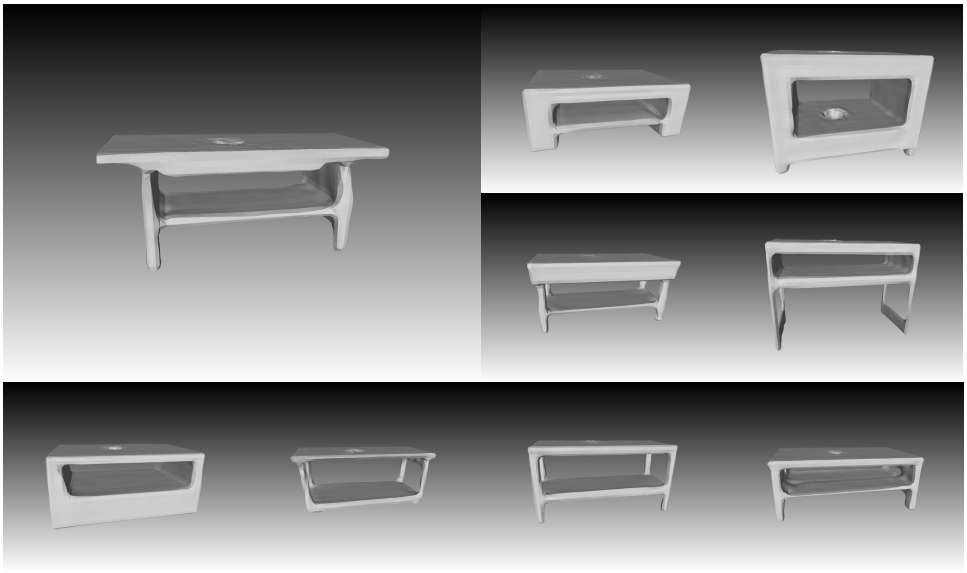
4.2 Text Conditioned Generation

Figure 4 中展示了各类文本条件生成的结果, 可以看到模型能够根据文本生成对应的物体形状, 并且能够控制一些生成的细节. 例如, 可以比较精细地控制椅子的腿部细节 (`chair with wheels` 和 `chair with base`), 与桌子的形状 (`an L shape table` 和 `a round table with several legs`). 同时生成的物体多样性较高, **Figure** 7 中展示了同一文本描述下丰富的生成结果. 在实际测试中也有一些失败的生成结果, 如 `chair with little pillow` 很难生成出小枕头的细节, 主要原因猜测是训练数据中相关的描述较少, 同时由于时间算力原因, 使用了中等大小的模型, 可能无法捕捉到所有细节.

Figure 6: Failure cases of the unconditional generation.



(a) "a somewhat circular chair"



(b) "a two-layer table"

Figure 7: More conditional 3D generation results of table and chair with text prompts.

5 Conclusion

本项目实现了基于 VQ-VAE 和 Diffusion Model 的 3D 物体生成, 包括无条件生成与文本条件生成. 目前得到的结果能够生成较为多样的物体形状, 在实验过程中发现相较于纯 `ResidualUNet3D` 模型, 添加了 Attention 模块后, 生成的物体的稳定性相对有所提升.
限于时间关系, 本项目没有尝试尝试实现 image-to-3D 的生成 (实际感觉上和文本条件生成类似, 使用一个预训练的图像编码器后, 训练 U-Net 即可). 同时没来得及尝试 texture generation/style transfer 的部分, 可以在之后的空闲时间中继续探索.
此外比较奇怪的是, 3D Diffusion Model 的开源代码尚未像 image Diffusion Model 有一个统一且活跃的代码库 (`diffusers`), 也许是我了解的不多.

Acknowledgements

感谢王鹏帅老师和助教们一学期的付出, 这是我所上的第一门图形学/三维视觉相关课程, 受益匪浅. 感谢 SDFusion, pytorch-3dunet, diffusers 等开源项目的作者们和社区贡献者们, 此外也感谢 VS Code Copilot(Claude Sonnet 4) 陪伴我一起 coding and debug :<).

References

[1] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, 2015.

[2] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. *arXiv preprint arXiv:1803.08495*, 2018.

[3] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tuyakov, Alex Schwing, and Liangyan Gui. SDFusion: Multimodal 3d shape completion, reconstruction, and generation. In *CVPR*, 2023.

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020.

[5] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[6] Adrian Wolny, Lorenzo Cerrone, Athul Vijayan, Rachele Tofanelli, Amaya Vilches Barro, Marion Louveaux, Christian Wenzl, Sören Strauss, David Wilson-Sánchez, Rena Lymbouridou, Susanne S Steigleder, Constantin Pape, Alberto Bailoni, Salva Duran-Nebreda, George W Bassel, Jan U Lohmann, Miltos Tsiantis, Fred A Hamprecht, Kay Schneitz, Alexis Maizel, and Anna Kreshuk. Accurate and versatile 3d segmentation of plant tissues at cellular resolution. *eLife*, 2020.