

大模型驱动三维资产程序化生成与评估方法

杭朋洁¹⁾, 卢奕南²⁾, 白桐源³⁾, 伍铁如^{3,4)}, 马锐^{3,4)*}

¹⁾ (吉林大学软件学院 长春 130012)

²⁾ (吉林大学计算机科学与技术学院 长春 130012)

³⁾ (吉林大学人工智能学院 长春 130012)

⁴⁾ (知识驱动人机智能教育部工程研究中心 长春 130012)
(ruim@jlu.edu.cn)

摘 要: 三维内容的自动化生成对于虚拟现实、游戏、影视等具有重要意义。针对根据自然语言指令高效、可控地生成高质量三维物体中存在的问题, 提出基于大语言模型的三维资产程序化生成与评估方法。首先将用户自然语言描述转换为 InfiniGen 库的脚本指令, 生成符合语义要求的三维模型, 实现从文本到代码再到三维物体的跨模态转换; 然后使用 CLIP 模型的筛选优化机制对生成结果进行评估和打分; 最后根据分数自动选择三维模型。实验结果表明, 与不采用 CLIP 筛选的基线方法相比, 所提方法显著地提高了生成三维模型的质量以及与输入文本的一致性; 在多个类别的三维物体生成任务上, 该方法取得了更高的 CLIP 匹配分数; 证明了将大模型与程序化三维生成技术相结合的可行性, 为跨模态可控生成三维内容提供了一条有效的路径。

关键词: 三维生成; InfiniGen; 大语言模型; 程序化

中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2025-00373

A Framework for Procedural Generation and Evaluation of 3D Assets Driven by Large Language Models

Hang Pengjie¹⁾, Lu Yinan²⁾, Bai Tongyuan³⁾, Wu Tieru^{3,4)}, and Ma Rui^{3,4)*}

¹⁾ (School of Software, Jilin University, Changchun 130012)

²⁾ (School of Computer Science and Technology, Jilin University, Changchun 130012)

³⁾ (School of Artificial Intelligence, Jilin University, Changchun 130012)

⁴⁾ (Engineering Research Center of Knowledge-Driven Human-Machine Intelligence, Changchun 130012)

Abstract: The automatic generation of 3D content is of great significance for virtual reality, games, film and television and other fields. In view of the problems existing in the efficient and controllable generation of high-quality three-dimensional objects according to natural language instructions, a method of programmed generation and evaluation of three-dimensional assets based on large language model is proposed. Firstly, the user's natural language description is transformed into the script instructions of InfiniGen library to generate a three-dimensional model that meets the semantic requirements, and realize the cross modal transformation from text to code and then to three-dimensional objects. Then, the screening optimization mechanism of clip model is used to evaluate and score the generated results. Finally, the 3D model is automatically selected according to the score. Experimental results show that compared with the baseline method without clip filtering,

收稿日期: 2025-09-10; 修回日期: 2025-12-08。基金项目: 国家自然科学基金 (62572212, 62202199); 吉林省自然科学基金 (20230101071JC)。杭朋洁 (1999—), 男, 硕士研究生, 主要研究方向为生成式人工智能; 卢奕南 (1969—), 女, 博士, 教授, 主要研究方向为图像处理、数据挖掘、生物信息处理; 白桐源 (1996—), 男, 博士研究生, 主要研究方向为三维场景生成; 伍铁如 (1971—), 男, 博士, 教授, 博士生导师, CCF 会员, 主要研究方向为 CAGD&CG、数值逼近、机器学习及其应用、工业软件; 马锐 (1988—), 男, 博士, 研究员, 博士生导师, CCF 会员, 论文通信作者, 主要研究方向为计算机图形学、计算机视觉、生成式人工智能、多模态大模型。

the proposed method significantly improves the quality of 3D model generation and the consistency with the input text. This method achieves higher clip matching scores for multi class 3D object generation tasks. The feasibility of combining large model with programmed 3D generation technology is proved, which provides an effective path for the controllable generation of 3D content across modes.

Key words: 3D generation; InfiniGen; large language models (LLM); programmatic

随着元宇宙、虚拟现实、电子游戏等领域的兴起,工业界对海量高质量三维数字资产的需求日益增长。然而,传统的三维模型创建依赖专业设计师使用复杂的建模软件手工构建,过程耗时且需要大量专业技能,这种高成本的内容生产方式难以满足日益增长的个性化和即时生成需求。近年来,基于深度学习的三维内容生成技术受到广泛关注。一些工作利用预训练的扩散模型和 NeRF^[1]场景表示,从文本描述中优化生成三维模型,如 DreamFusion^[2],该方法证明了通过二维图像生成模型可以实现逼真的文本到三维合成,但存在优化耗时长(通常需要数小时)且结果分辨率受限等问题;之后, Magic3D^[3]通过两阶段优化将生成时间缩短至约 40 min,并提高了最终三维网格的清晰度。尽管基于生成模型的方法能够在零样本条件下合成任意类别的三维形状,但是计算成本和对硬件的要求依然较高,且对生成结果的精确可控性不足。

程序化三维内容生成利用参数化建模和规则系统快速地生成场景和物体。InfiniGen 库^[4-5]基于一套预定义的过程规则,在 Blender^[6]中合成地形、植被、家具等丰富的三维资产,无需学习并且在无 AI 参与、仅使用纯算法规则的情况下,即可生成多样且逼真的三维资产,已经被用于大规模合成训练数据。然而,纯程序化的方法通常缺乏高级语义层面的控制,用户需要了解底层生成库的参数和调用方式才能定制想要的结果,对于非专业用户的门槛较高,限制了程序化生成在内容创作中的灵活应用。

当前,大语言模型(large language model, LLM)^[7-8]在理解和执行复杂指令方面展示出强大的能力,有望成为连接自然语言与专业工具的桥梁。研究人员开始探索利用 LLM 辅助三维内容创作。Sun 等^[9]提出的 3D-GPT 框架将 LLM 用作任务规划者,将复杂的三维建模任务分解并调度给不同模块执行,该框架集成了任务分发、概念生成和建模 3 个智能代理,从简要文本逐步细化场景描述,并提取参数生成 Blender 可执行脚本,创建目标三维场景。此外,一些工作聚焦于将 LLM 用于三维编辑。Huang 等^[10]提出 BlenderAlchemy 系统,

利用视觉语言模型(如 GPT-4V)结合“编辑生成器+状态评估器”进行迭代搜索,在 Blender 中自动执行序列化编辑操作,以满足用户指定的目标效果,通过让视觉语言模型感知中间渲染结果并进行调整,实现了材质、几何、灯光等复杂编辑的自动化。还有的研究致力于提升 LLM 生成三维代码的准确性。Du 等^[11]提出 BlenderLLM 框架,构建了包含约 12k 指令-脚本对的 BlendNet 数据集,采用少样本微调和自我改进训练,极大地提高了 LLM 生成 CAD 脚本的功能性和正确率。

上述工作表明,将大模型与三维引擎/工具结合,能够在一定程度上实现从文本到三维内容的转化;然而,目前的方法仍有诸多不足,例如,对生成结果缺乏自动的质量评价和反馈,需要人工挑选满意结果;在复杂物体细节和语义一致性上有所欠缺。为了进一步推动大模型辅助三维生成技术的发展,实现更加自动化、高质量和可控的文本生成三维物体,本文提出大模型驱动三维资产程序化生成与评估方法。

首先设计一个基于 LLM 的三维物体生成框架,利用 LLM 理解用户的自然语言指令,并生成调用 InfiniGen 库的 Python 脚本代码,自动合成符合指令语义的三维模型;该框架打通“文本描述→代码表示→三维对象”的跨模态转化通道,大幅降低了用户生成定制三维内容的技术门槛。

然后提出多样化生成及筛选机制,利用 InfiniGen 自带的随机化参数保证生成结果的多样性,引入 CLIP 多模态模型^[12]对每轮生成的结果进行图文相似度评估,根据 CLIP 得分自动筛选并保留最符合指令的结果。

最后,通过定量和定性实验验证本文方法的有效性以及各模块的作用;与不同基线方法相比,该方法取得了更加符合文本语义一致性和高质量的结果。

1 相关工作

1.1 基于生成式模型的三维资产生成

随着文本到图像生成建模研究的不断深入,已

有一些文本到三维模型生成的方法^[2,13-18],它们都是基于可微渲染的方式对每个形状优化,同时通过 CLIP 模型或 2D 扩散模型^[19]提供的指导信号进行联合训练。DreamFields^[17]和 CLIP-Mesh^[16]使用 CLIP 实现了零样本三维内容创建; DreamFusion^[2]在文本到图像扩散模型的指导下优化 NeRF^[1],实现了良好的文本到三维合成结果;为了应对优化速度和视觉质量问题挑战, Magic3D^[3]使用低分辨率扩散先验和稀疏三维哈希网格保证速度优势,同时运用高效可微渲染技术对纹理三维网格模型进行精细优化,在速度提升和视觉质量改善之间实现了平衡。为了提高生成的三维模型的保真度,许多方法聚焦于改进三维几何形状。Dream3D^[14]使用生成的符号距离函数(signed distance function, SDF)直接初始化 NeRF,实现更好的几何控制; Latent-NeRF^[15]则将用户提供的网格与 NeRF 优化相结合,引入直接占用损失进一步优化几何结构。此外,主题驱动的文本到三维生成方法在个性化合成方面也取得了一定的进展,该方法根据输入图像和文本提示创建特定主题的三维资产。为了实现这一目标, Raj 等^[20]提出一种三阶段优化策略 DreamBooth3D,融合了 NeRF 的三维一致性以及文本到图像扩散模型的个性化能力; One-2-3-45^[21]则借助 Zero123 这种多视图 2D 扩散模型生成多视图图像,用于学习基于 SDF 的可泛化神经表面重建。尽管上述方法在文本到三维模型转化方面取得了一定进展,但是它们往往难以精准地把握文本语义,在描述物体细节时容易出现语义偏差。为了有效地解决这一问题,本文方法引入文本理解智能体,利用 LLM 对文本信息进行扩充,在一定程度上克服了语义偏差的难题。

1.2 基于大模型的三维资产生成

LLM 作为一种创新方法,能够有效地捕捉并压缩人类的知识与经验,并将其映射到语言空间^[7,22]。近年来,LLM 在传统自然语言处理领域之外也展现出了强大的泛化能力,其中,基于 LLM 的三维资产生成技术取得了突破性的进展。在几何表征建模方面, MeshGPT^[23]创新性地仅基于 Transformer 的解码器架构,通过自回归序列预测实现了三角网格的端到端生成,其网格拓扑优化方法显著地提升了生成质量;进一步, LLaMA-Mesh^[24]将语言模型与几何建模相融合,通过文本语义指导下的潜在空间映射建立了语言指令与三维形态的关联通道; ShapeGPT^[25]构建统一的多模态语言模型框架,通过跨模态注意力机制实现了文本、点云、体素等多

维度数据的联合建模,与传统方法相比,其形状补全任务准确率有较大的提升; DreamLLM^[26]通过协同多模态理解-生成架构,在保持几何一致性的同时,实现了跨文本、图像、三维数据的双向推理能力。针对工程领域特殊需求, CadVLM^[27]创新性地引入视觉-语言模型引入参数化 CAD 建模,其基于草图语义解析的生成方法在机械设计基准测试中达到了较高的水准。然而,现有方法在生成模型的质量评估方面存在不足,通常需要人工介入挑选满意结果,无疑增加了人力成本且效率低下。为了解决这一问题,本文方法利用 InfiniGen 生成多个三维模型,在指定视角下渲染后,再用 CLIP 模型自动筛选出与文本语义一致的结果,减少人工参与,有效地提升了效率。

1.3 程序化的三维资产生成

近年来,程序化三维资产生成技术作为三维内容生产的重要分支展现出独特的优势,其核心是依据预设规则与参数算法自动生成复杂结构模型。传统的程序化方法包括基于 L-System 的植物生长建模、基于 Grammar 的建筑和城市建模等,这些方法虽然能够通过少量参数生成丰富变化的结构,但是通常需要人工设计规则。InfiniGen^[4]是最近兴起的一个通用程序化三维资产生成库,在不使用深度学习的情况下预定义了一系列生成器,可以在 Blender 中自动创建地形、植物、家具等多种复杂场景。程序化方法生成速度快且结果逼真,弱点是没有语义理解能力,用户需要通过手动设置参数或编写脚本实现想要的效果,无法直接用自然语言描述目标并得到响应。

结合大模型的语义理解能力与程序化引擎的生成能力,是实现高效可控三维内容生成的有前景方向,但是,现有方法在生成质量控制、物体细节等方面仍有提升空间。在此基础上,本文引入概念增强、多样化生成和 CLIP 筛选机制,进一步提高生成的语义准确性和视觉质量。

2 本文方法

本文提出的大模型驱动的三维资产生成方法,架构如图 1 所示,由文本理解智能体、代码生成智能体、三维模型生成和结果评估筛选 4 个模块组成,以及一个集成 LLM、程序化生成库和多模态评估的系统。

算法 1: 文本到三维模型生成算法。

输入: 文本指令 T 。

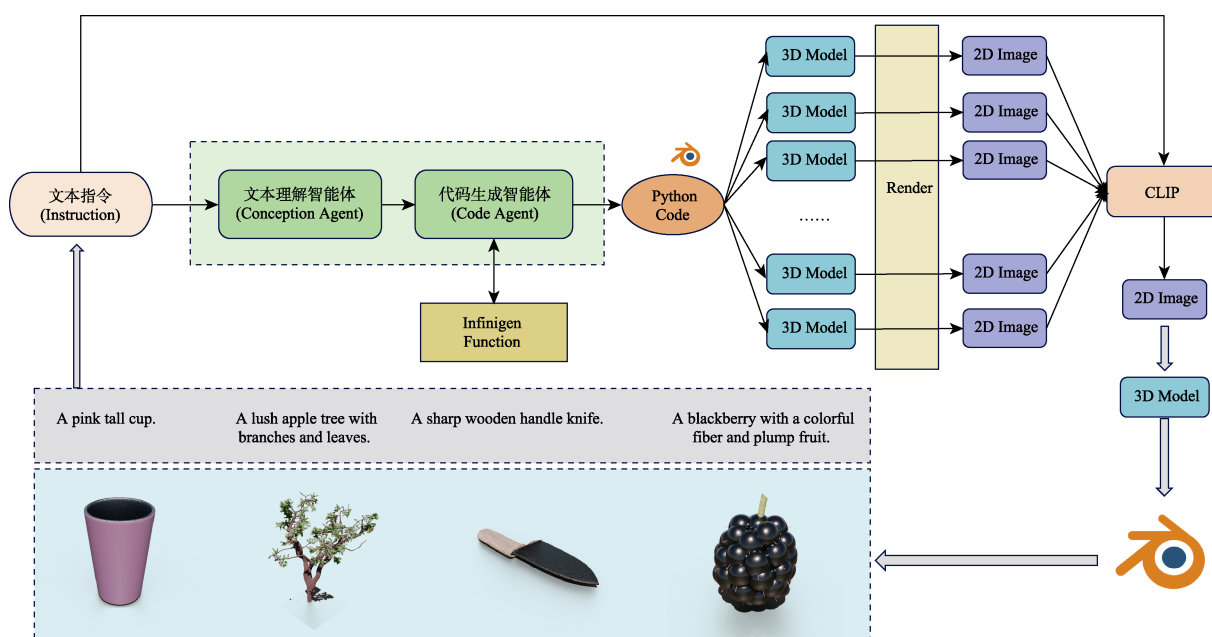


图1 本文方法架构

输出: 三维模型 M 。

$C \leftarrow \text{ConceptionAgent}(T)$ // 文本理解, 得到概念描述

$P \leftarrow \text{CodeAgent}(C)$ // 代码生成, 得到 Python 脚本

$\text{Models} \leftarrow \emptyset$

for each $p \in P$ do

$m \leftarrow \text{Blender.Execute}(p)$ // 调用 Blender/Infinigen

生成三维模型

$\text{img} \leftarrow \text{Blender.Render}(m)$ // 渲染得到 2D 图像

$\text{score} \leftarrow \text{CLIP}(T, \text{img})$ // 计算文本-图像一致性

$\text{Models} \leftarrow \text{Models} \cup \{(m, \text{score})\}$ // 选择一致性

最高的模型

end for

$M \leftarrow \text{argmax}_{\{m \in \text{Models}\}} \text{score}$

return M

2.1 大模型驱动的三维模型程序化生成

(1) 文本理解智能体

在文本理解阶段, 采用 GPT-4^[7] 扩充用户的文本描述, 利用大模型的先验和推理能力更详细地描述用户的意图; 然后将扩充的文本描述与原有文本描述相结合作为新的文本描述。本文针对不同的物体设计不同的提示模板生成符合用户意图的文本描述。文本理解智能体提示模板的结构如图 2 所示。其中, LLM System 部分定义了为 LLM 设定的角色; User 部分的 INSTRUCTION 是用户的原始文本描述, INFO 明确了 LLM 在对物体进行详细描述时需要重点关注的信息类别, 不同物体的 INFO 有不同的描述, Task 则用于输入文本理解智能体, 引导其进行准确且细致的文本扩充和语义解析任务, 为后续代码生成及三维模型构建提供精准的语义基础和详细的指导信息。

-LLM System: You are a skilled and wise writer, skilled in describing objects, especially their appearance and various details.
-User: INSTRUCTION
-Task: Given the user's text description INSTRUCTION, please provide a detailed description of the following information: INFO. For the content not mentioned in the description, please use your imagination to ensure that it conforms to the laws of the physical world and the textual description.

图2 文本理解智能体提示模板的结构

(2) 代码生成智能体

在代码生成阶段, LLM 将新的文本描述翻译为可执行的 Blender Python 脚本, 通过设计专门的提示模板, 引导 LLM 输出既符合语法规则、又能准确生成目标物体的脚本。代码生成智能体提示模板的结构如图 3 所示。其中, LLM System 部分明确

-LLM System: You are a smart and excellent 3D designer, proficient in using Python, able to convert natural language descriptions into Python parameters, and understand the use of Python functions to generate 3D content.
-Conception Agent: DESCRIPTION
-Task: Given a text description: DESCRIPTION, we have the following function code FUNCTION, as well as the documentation for this function DOCUMENT. Here is an example on how to model an object using function calls to make it more in line with the text description: EXAMPLE. Please understand this function and model the 3D object described in the text by calling it.

图3 代码生成智能体提示模板的结构

定义了为 LLM 设定的角色; Conception Agent 部分中, *DESCRIPTION* 融合了用户的原始文本描述和文本理解智能体的输出结果; Task 部分中, *FUNCTION* 指定用于生成三维物体的 Python 函数, *DOCUMENT* 提供该 *FUNCTION* 函数的详细文档说明, *EXAMPLE* 则为代码智能体提供相应的示例, 以便更好地理解 and 生成符合要求的脚本。

用户首先使用自然语言描述其期望的三维物体; 然后文本理解智能体负责解析该指令的语义, 并基于知识库对该指令进行语义扩充; 代码生成智能体依据内置知识和相关提示生成相应的 Python 脚本代码, 该过程运用 InfiniGen 提供的程序化生成库, 通过调用库中预设的植物、家具、地形等生成器构建所需的三维物体模型。例如, 当用户给出文本描述 “A pink tall cup.” 时, 本文系统生成一个三维模型脚本代码, 用于创建一个粉色高杯的三维模型, 文本理解智能体的输出结果如图 4 所示, 即 Conception Agent Output。初始文本描述和文本理解智能体的输出作为代码智能体的输入, 代码智能体的执行结果如图 5 所示。

2.2 多样化三维资产生成与评估

(1) 多样化生成

在三维模型生成阶段, 通过代码智能体输出的

代码、InfiniGen 库和 Blender 创建三维模型。为了实现多样化生成, 本文取消固定随机种子的设置, 并利用 InfiniGen 库自带的随机参数功能, 通过指定生成的三维模型数量, 可以生成任意多个不同的三维模型, 这些模型在整体结构上保持一致, 但在细节上各有差异。本文使用文本描述 “This is a tree with lush branches and leaves, with a straight trunk and branches extending outwards. The leaves appear bright green, and the tree is covered with red fruits, giving it a vibrant and lively appearance.” 生成了 45 个细节各异的树木模型, 如图 6 所示。

(2) 基于 CLIP 的资产评估

在三维模型生成后, 需要确定哪些模型更符合用户的期望。在缺乏真实参考的情况下, 传统方法通常依赖人工检查, 无疑增加了工作的难度。为了解决这一问题, 实现无人监督的结果评估, 本文提出基于 CLIP 的三维资产评估模型, 对三维生成的结果和输入文本的语义一致性进行评估。CLIP 由 OpenAI 提出, 已经被证明能够在无参考的情况下有效地判断图文相关性, 其评估结果与人类评价具有较高的一致性。将每个三维模型渲染后得到的图像 I 与用户的原始文本描述 T 一起输入 CLIP 模型, 获得相似度得分 $\text{Sim}(I, T)$, 即 CLIP Score, 其值越高, 说明图像内容与文本语义的匹配度越高。

INSTRUCTION: A pink tall cup.

INFO: If the given text is describe the cup, provide whether the cup is a short cup, whether the contour is straight, the lowest coordinate value of the cup width, the depth of the cup, and whether the cup is protected. If these are not provided, use your imagination based on the existing scene.

-LLM System: You are a skilled and wise writer, skilled in describing objects, especially their appearance and various details.

-User: **INSTRUCTION**

-Task: Given the user's text description **INSTRUCTION**, please provide a detailed description of the following information: **INFO**. For the content not mentioned in the description, please use your imagination to ensure that it conforms to the laws of the physical world and the textual description.

-Conception Agent Output: This cup is one of pride and elegance, standing not as a short, unassuming object but rather a towering porcelain spectacle further accentuated by its dynamic color palette. It carries a delightfully mesmerizing shade of pink, the kind that teeters between soothing pastel and invigoratingly vivid. This hue exudes an aura of innocence and cheer, adding a gentle dash of charm to anywhere it's placed. It feels like the embodiment of a warm spring day right at the break of dawn when the sky baskers in a similar blend of rosy illumination. As for the shape, the cup departs from the conventional curves, opting instead for a linear contour, unyieldingly straight from top to bottom. It presents a streamlined silhouette that borders on minimalistic design, the cylindrical form striking a balance between practicality and aesthetic simplicity. The cup, despite its towering height, manages to maintain a delicate equilibrium in terms of its width. It's thin, yet substantial enough to provide a secure grip. The lower limit of the cup's diameter, one might assume, is at a modest estimate around 6 centimeters, providing enough room for the liquid but also ensuring easy maneuverability. Underneath the poised exterior, the depth of the cup is equally impressive, matching its height with careful symmetry. As a tall cup, it wouldn't be a far stretch to visualize the depth nearing 8 to 10 centimeters. One can surely envision enjoying a generous serving of their favorite beverage in such a vessel. As for protection, the cup boasts a robust build despite its refined presence. While the text doesn't explicitly mention any protective cover or features, it's easy to assume that the cup carries a smooth, matte finish, significantly reducing the risk of slippage. And while it might not sport a protective silicone base or a handle to minimize heat transfer, a cup of its stature could likely endure typical wear and tear, serving as a reliable companion for countless cozy evenings to come. To further its longevity, it could have heat resistant characteristics and a decent resistance against chipping or cracking. With its grand structure and delightful coloring, the pink tall cup emerges as a beacon of inviting warmth and comforting functionality.

图 4 文本理解智能体输出示例

DESCRIPTION
FUNCTION:
`import CupFactory`
`def add_cup(scene, is_short, is_profile_straight, x_lowest, depth, has_guard):`
`cup = CupFactory.create(is_short, is_profile_straight, x_lowest, depth, has_guard) # create cup`
`scene.placement(cup) # place cup to the scene`
`return`

DOCUMENT:
Explanation: The function takes built indoor scene as input, it will add cup to the scene.
Inputs:
scene: the built indoor scene.
is_short (bool): Indicates if the cup is short. True for a short cup, False for a taller one.
is_profile_straight (bool): Determines if the cup's profile is straight. True for straight, False for a curved profile.
x_lowest (float): The lowest x-coordinate value defining the cup's width, ranging from 0.6 to 1.0.
depth (float): The depth of the cup, ranging from 0.25 to 1.0.
has_guard (bool): Indicates if the cup has a guard. True to include a guard, False otherwise.

EXAMPLE:
Question: given the text description of the scene: "A red teacup." analysis the function parameter and call the function to add cups that fits the text description.
Solution: We can identify one red cup in the scene from the text description. We know that tea cups are usually low footed cups, so we set is_short to True. The outline of the teacup may or may not be straight. We can randomly choose True or False, and here we set is_profile_straight to True. The width of the teacup is not very wide, we set x_lowest to 0.7. The depth of the teacup is not very deep, we set it to 0.3. Tea cups are usually protected, so we set has_guard to True.
We can call the function one time to create a teacup: ``python add\_cup(scene, True, True, 0.7, 0.3, True) # create teacup``

-LLM System: You are a smart and excellent 3D designer, proficient in using Python, able to convert natural language descriptions into Python parameters, and understand the use of Python functions to generate 3D content.
-Conception Agent: *DESCRIPTION*
-Task: Given a text description: *DESCRIPTION*, we have the following function code *FUNCTION*, as well as the documentation for this function *DOCUMENT*. Here is an example on how to model an object using function calls to make it more in line with the text description: *EXAMPLE*. Please understand this function and model the 3D object described in the text by calling it.

Code Agent Output: Given the text description, we can identify that the cup is tall, has a straight profile, has a width of around 6 centimeters (which we can normalize to 0.6 in the function parameters), has a depth of around 8 to 10 centimeters (which we can normalize to 0.8 or 0.9), and does not explicitly mention any protective guard. So, we can set is_short to False, is_profile_straight to True, x_lowest to 0.6, depth to 0.8 or 0.9, and has_guard to False. We can call the function once to create the cup as follows: ``python add\_cup(scene, False, True, 0.6, 0.9, False) # create pink tall cup``

图 5 代码智能体输出示例



图 6 基于同一指令生成的多样化结果

从多个结果对应的渲染图像中, 本文挑选出与文本描述最对应的图像以及相应的三维模型, 即 CLIP Score 最高的模型作为最终结果。通过这种方法, 能够有效地过滤出语义正确且视觉效果自然的模型。

3 实验及结果分析

3.1 实验设置

本文系统以 LLM 作为核心驱动力, LLM 负责将文本转化为生成代码, InfiniGen 在三维模型生成过程中引入随机参数以增加结果的多样性; Blender 担当后端引擎, 负责实际生成三维模型及渲染图像; CLIP 则作为质量监督信号, 确保生成内容与用户期望的关联性和准确性。整个流程高度自动化, 用户只需要输入指令并设定初始参数, 系统便能自主运行, 最终输出令人满意的三维物体。本文选用在理解复杂指令和生成代码方面表现出色的 GPT-4 作为主要语言模型, InfiniGen 使用其 github 网站上公布的开源代码, CLIP 采用 OpenAI 提供的 ViT-B/32 模型。计算图文相似度时, 本文将用户指令作为文本输入, 在指定视角范围内随机选取一个视角对生成的物体进行渲染, 并将渲染的图像作为输入。

为了评估本文方法的性能, 本文设计了一系列定量与定性实验。在测试数据方面, 鉴于目前缺乏公开的文本到三维物体生成基准, 本文自主构建了一个包含 50 条文本指令的测试数据集, 涵盖自然类、室内物品类等多种类别以及不同难度的描述。所有指令均由 GPT-4 生成, 采用如图 7 所示的生成数据提示模板, 其中, *OBJECT* 表示某一特定物体, 测试数据集中的部分指令及其对应的三维模型示例如图 8 所示; 所有实验均基于这一指令集合进行, 确保实验结果的可比性和一致性。目前已有的若干公开的文本到三维生成基准, 如 GT23D-Bench、Text-to-Shape Benchmark 等, 主要针对基于扩散模型或 NeRF 的生成式方法设计, 强调几何一致性和多视图渲染质量。本文方法基于程序化生成 (InfiniGen

-LLM System: You are a clever and sharp writer, skilled at describing objects in short language.
-User: *OBJECT*
-Task: Given the object category *OBJECT*, please use your imagination to describe the object in concise words, ensuring that it conforms to the laws of the physical world.

图 7 生成数据提示模板

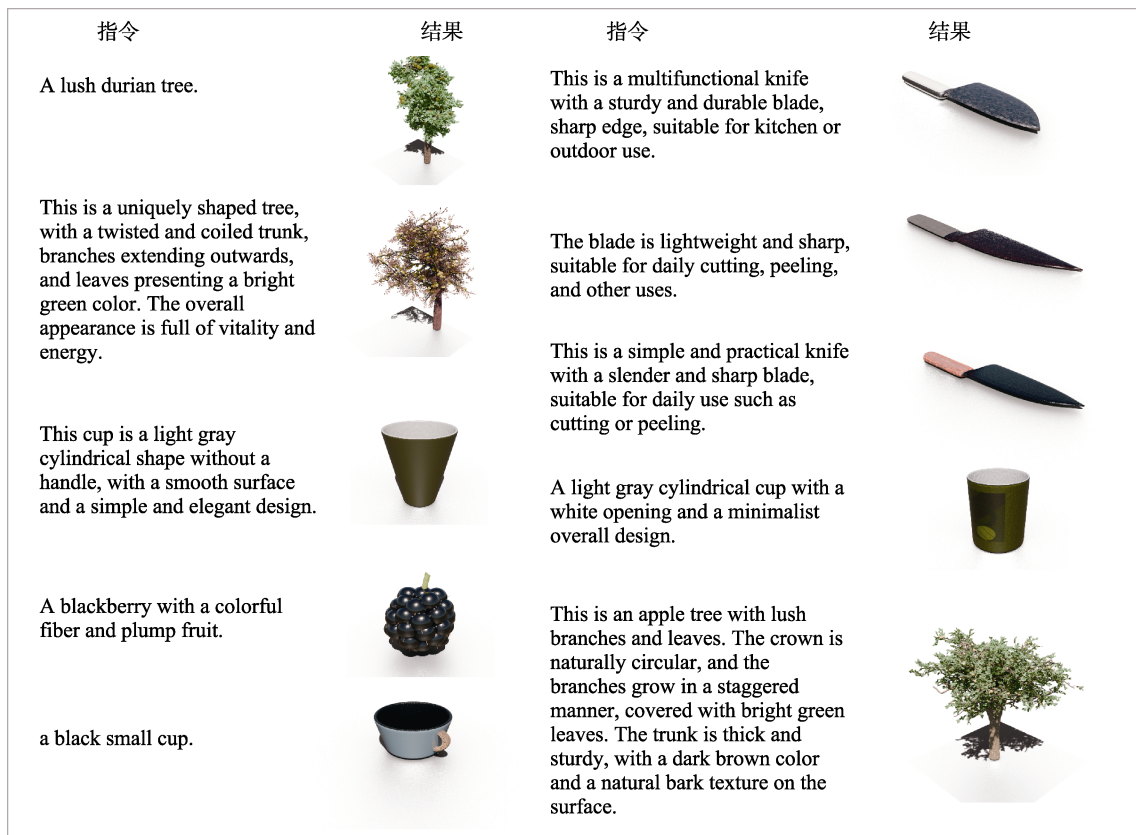


图 8 测试数据集中的部分指令及其对应的三维模型示例

库), 支持的物体类别(如植物、家具)与这些基准的部分重叠有限, 且评估侧重于程序化效率和语义匹配。为了确保与 InfiniGen 的兼容性, 本文选择自定义数据集进行实验。未来, 计划扩展到 GT23D-Bench 等基准上进行进一步验证, 以更全面地评估方法的泛化性。

本文采用 CLIP Score 作为自动评价指标, 通过 CLIP 模型计算生成的三维模型在指定视角下的渲染图像与对应文本的相似性分数, 能够在一定程度上有效地反映生成模型与文本语义的匹配程度、语义相关性和内容的准确度。

本文所有实验均在配置为 Intel i7-13700KF CPU、NVIDIA RTX 3060 GPU, 以及 Intel i7-13650HX CPU、NVIDIA RTX 4060 GPU 的 2 台机器上运行。在模型生成效率方面, 不同类型的三维模型生成时间存在差异, 通常, 面数较多的模型生成时间更长, 使用 Blender/InfiniGen 生成单个三维模型的平均耗时约为 1~2 min (包含渲染时间)。在 GPT-4、Llama-3-8B-Instruct、Qwen-14B 和 DeepSeek-R1 这 4 种 LLM 下, 模型生成时间对比如表 1 所示, 本文在 2 台机器上同时生成模型, 每台机器各自承担 50% 的任务。可以看出, CLIP 评估几乎可以实时完成, 主要耗时环节为生成三维模型。

表 1 不同 LLM 下生成时间对比

LLM	时间/s
GPT-4 ^[7]	4.65
Llama-3-8B-Instruct ^[28]	1.92
Qwen-14B ^[29]	2.32
DeepSeek-R1 ^[8]	9.16

3.2 消融实验

由于目前缺乏将 LLM 和 InfiniGen 直接结合进行单个三维资产生成的框架, 为了验证本文方法及其各模块的有效性, 通过设置 3 组消融实验, 并将其作为基线模型, 与本文方法进行定量和定性的对比。消融实验设置如下:

(1) No-Con-CLIP: 不使用文本理解智能体也不使用 CLIP 评估, 相当于使用大模型直接生成 InfiniGen 可以接受的脚本; 其中, 不使用 CLIP 评估指只使用一组参数生成一个三维模型。本文将 No-Con-CLIP 作为最基本的基线模型。

(2) No-CLIP: 不使用 CLIP 评估, 即只使用一组参数生成一个三维模型。

(3) No-Con: 使用 CLIP 评估, 但不使用文本理解智能体。

在 50 条测试指令的测试数据集上, 消融实验结果如表 2 所示。可以看出, 本文方法成功地生成了与文本语义高度匹配的三维模型, CLIP Score 平均达到 28.924 6, 显著优于不使用筛选优化的 No-CLIP 方案, 表明优化筛选在提升文本语义对齐方面具有显著效果; 与 No-Con 相比, 该方法的 CLIP Score 高出约 0.72, 表明文本理解智能体有助于代码生成智能体更准确地理解用户意图, 从而提高生成结果的文本语义对齐程度; No-Con-CLIP 的 CLIP Score 最低。在生成效率方面, 与 No-Con-CLIP 和 No-Con 相比, 本文方法仅多出一次 LLM 响应时间, 与 No-CLIP 的生成速度几乎持平。

表 2 在测试数据集上的消融实验结果对比

模型	CLIP Score			
	GPT-4 ^[7]	Llama-3-8B-Instruct ^[28]	Qwen-14B ^[29]	DeepSeek-R1 ^[8]
No-Con-CLIP	24.1604	16.0945	18.0825	23.1602
No-CLIP	24.3339	16.3565	18.8565	25.1702
No-Con	28.2076	19.0982	22.9632	27.5632
本文	28.9246	19.5056	23.1023	27.9656

注: 粗体表示最优值。

在多条指令下, 消融实验生成结果的可视化对比如表 3 所示。可以看出: (1) 在第 1 条指令下, No-Con-CLIP 生成的树叶颜色与树干粗度的维度存在显著语义偏移; No-CLIP 生成的树枝分布杂乱且树干纤细; No-Con 虽然实现了枝条交错生长, 但是叶片稀疏、树干过细; 采用本文方法处理后, 最终生成的树冠层次分明, 枝干粗壮自然。(2) 在第 2 条指令下, No-Con-CLIP 错误地将刀体材质建模为木质结构, 与文本要求的“锋利”属性矛盾; No-CLIP 生成刀具的刃口曲率显著偏离锐利特征; No-Con 在色彩还原度之外的设计细节与本文方法差异较小, 表明对于具有明确特征描述的复杂指令, 即便未经过文本理解智能体处理仍具备基础语义表达能力。(3) 在第 3 条指令下, No-Con-CLIP、No-CLIP 和 No-Con 均出现显著的特征偏移现象; 本文方法在简单指令场景下展现出更精确的语义捕捉能力。通过实验, 验证了不同模型在复杂和简单指令场景下的性能边界, 凸显了本文方法在跨尺度特征建模方面的技术优势。

消融实验的定量和定性结果均表明, 本文提出的各模块在提升生成质量上各有贡献: 文本理解智能体作为核心毫无疑问是不可或缺的, 缺少它将无法理解指令中的复杂语义; CLIP 筛选更符合文本语

表 3 在 4 条典型指令下消融实验生成结果的可视化对比

编号	指令	本文	No-Con	No-CLIP	No-Con-CLIP
1	This is an apple tree with lush branches and leaves. The crown is naturally circular, and the branches grow in a staggered manner, covered with bright green leaves. The trunk is thick and sturdy, with a dark brown color and a natural bark texture on the surface.				
2	The blade is lightweight and sharp, suitable for daily cutting, peeling and other operations, and easy to carry.				
3	This is a beige yellow cylindrical cup with a black rim.				
4	A lush durian tree.				

义的结果；相比之下，如果只用 LLM 生成而无 CLIP 筛选，生成的成功率和准确率都将大打折扣。

3.3 对比实验

在 50 条测试数据集上，将本文方法与代表性的文本到三维生成方法进行实验，包括基于生成式模型的 DreamFusion^[2]和 Magic3D^[3]，以及基于大模型的 3D-GPT^[9]，进一步验证本文方法的优势，结果如表 4 所示。可以看出：（1）对比方法虽然能够生成三维模型，但是缺乏类似本文的 CLIP 自动筛选机制，在复杂物体细节（如纹理分布和几何精确性）和语义一致性上易出现偏差。（2）本文使用 CLIP Score 作为评估指标，对所有方法生成的模型进行评估（渲染后计算），本文方法的平均 CLIP Score 为 28.9246，显著高于对比方法，表明引入筛选机制能够有效地提升语义一致性；尤其在复杂描述（如树木枝叶细节）上，对比方法因无自动优化而欠缺。（3）在生成效率方面，本文方法生成每个模型平均耗时 1~2 min，而 DreamFusion 和 Magic3D 需要数十分钟，凸显了程序化+筛选的效率优势。

表 4 在测试数据集上 4 种方法的 CLIP Score 对比

方法	CLIP Score
DreamFusion ^[2]	25.3124
Magic3D ^[3]	26.7812
3D-GPT ^[9]	26.8816
本文	28.9246

注：粗体表示最优值。

4 结 语

本文提出大模型驱动的三维资产程序化生成与评估方法，以及一个集成 LLM、程序化生成库和多模态评估的系统。以 LLM 为桥梁，本文实现了从自然语言描述到三维模型的自动转换，用户只需给出文本指令即可生成符合语义的三维物体；引入 CLIP 模型对结果进行评价筛选，有效地提升了生成内容的语义准确度和视觉品质。通过实验，证实了本文方法在不同类别物体生成上的有效性，与传统的一次性生成或深度学习优化方法相比，具备

更高的可控性和效率。

本文研究表明, 将 LLM 与程序化三维引擎相结合是可行且高效的, 为数字内容创作提供了新范式, 能够降低三维设计门槛, 加速模型迭代, 对于需要海量定制三维资产的领域 (如游戏、影视、教育等) 具有潜在价值。本文生成的三维模型基于 Blender 格式, 在 Blender 内可以导出为 fbx 或者 obj 格式, fbx 或者 obj 格式可以直接导入 Unity 中使用。但是, 本文方法也存在一些局限: 生成结果的精细逼真度仍然不如通过学习模型得到的效果, 未来可以考虑融合生成对抗网络或扩散模型的细节补充能力, 进一步提高视觉质量; 单视角 CLIP 评估存在误差, 后续将使用多视角 CLIP 评估来避免偏差; 生成的三维资产种类有限, 主要取决于 InfiniGen 中的三维资产数量; 该方法主要利用预训练 LLM 的通用能力, 尚未针对三维场景/三维模型进行微调, 今后可以结合 BlendNet 等数据对 LLM 进行微调, 提升其专业水平, 生成更复杂的三维场景/三维模型; 大模型赋能三维内容生成仍然有广阔空间, 有望逐步发展成为新一代智能三维设计助手, 实现“所言即所得”的创作体验。

参考文献 (References):

- [1] Mildenhall B, Srinivasan P P, Tancik M, *et al.* NeRF: representing scenes as neural radiance fields for view synthesis[C] //Proceedings of the 16th European Conference on Computer Vision. Heidelberg: Springer, 2023: 405-421
- [2] Poole B, Jain A, Barron J T, *et al.* DreamFusion: text-to-3D using 2D diffusion[C] //Proceedings of the 11th International Conference on Learning Representations. ICLR, 2023: 1-20
- [3] Lin C H, Gao J, Tang L M, *et al.* Magic3D: high-resolution text-to-3D content creation[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2023: 300-309
- [4] Raistrick A, Lipson L, Ma Z Y, *et al.* Infinite photorealistic worlds using procedural generation[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2023: 12630-12641
- [5] Lee W, Lee J, Seo J, *et al.* InfiniGen: efficient generative inference of large language models with dynamic KV cache management[C] //Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation. Santa Clara: USENIX, 2024: 155-172
- [6] Blender. Blender: a 3D creation suite. Blender foundation[OL]. [2025-09-10]. <https://www.blender.org>
- [7] OpenAI. GPT-4 technical report[OL]. [2025-09-10]. <https://arxiv.org/abs/2303.08774>
- [8] DeepSeek-AI. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning[OL]. [2025-09-10]. <https://arxiv.org/abs/2501.12948>
- [9] Sun C Y, Han J L, Deng W J, *et al.* 3D-GPT: procedural 3D modeling with large language models[C] //Proceedings of the International Conference on 3D Vision. Los Alamitos: IEEE Computer Society Press, 2025: 1253-1263
- [10] Huang I, Yang G D, Guibas L J. BlenderAlchemy: editing 3D graphics with vision-language models[C] //Proceedings of the 18th European Conference on Computer Vision. Heidelberg: Springer, 2024: 297-314
- [11] Du Y H, Chen S N, Zan W B, *et al.* BlenderLLM: Training large language models for computer-aided design with self-improvement[OL]. [2025-09-10]. <https://arxiv.org/abs/2412.14203>
- [12] Radford A, Kim J W, Hallacy C, *et al.* Learning transferable visual models from natural language supervision[C] //Proceedings of the 38th International Conference on Machine Learning. New York: PMLR, 2021: 8748-8763
- [13] Sanghi A, Chu H, Lambourne J G, *et al.* CLIP-forge: towards zero-shot text-to-shape generation[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 18582-18592
- [14] Xu J L, Wang X T, Cheng W H, *et al.* Dream3D: zero-shot text-to-3D synthesis using 3D shape prior and text-to-image diffusion models[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2023: 20908-20918
- [15] Metzger G, Richardson E, Patashnik O, *et al.* Latent-NeRF for shape-guided generation of 3D shapes and textures[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2023: 12663-12673
- [16] Khalid N M, Xie T H, Belilovsky E, *et al.* CLIP-Mesh: generating textured meshes from text using pretrained image-text models[C] //Proceedings of the SIGGRAPH Asia Conference Papers. New York: Association for Computing Machinery, 2022: Article No.25
- [17] Jain A, Mildenhall B, Barron J T. Zero-shot text-guided object generation with dream fields[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 857-866
- [18] Wang Z Y, Lu C, Wang Y K, *et al.* ProlificDreamer: high-fidelity and diverse text-to-3D generation with variational score distillation[C] //Proceedings of the 37th International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc, 2023: Article No.368
- [19] Rombach R, Blattmann A, Lorenz D, *et al.* High-resolution image synthesis with latent diffusion models[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 10674-10685
- [20] Raj A, Kaza S, Poole B, *et al.* DreamBooth3D: subject-driven text-to-3D generation[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2023: 2349-2359
- [21] Liu M H, Xu C, Jin H A, *et al.* One-2-3-45: any single image

- to 3D mesh in 45 seconds without per-shape optimization[C] //Proceedings of the 37th International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc, 2023: Article No.976
- [22] Devlin J, Chang M W, Lee K, *et al.* BERT: pre-training of deep bidirectional transformers for language understanding[C] //Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Stroudsburg: Association for Computational Linguistics, 2019: 4171-4186
- [23] Siddiqui Y, Alliegro A, Artemov A, *et al.* MeshGPT: generating triangle meshes with decoder-only transformers[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2024: 19615-19625
- [24] Wang Z Y, Lorraine J, Wang Y K, *et al.* LLaMA-Mesh: unifying 3D mesh generation with language models[OL]. [2025-09-10]. <https://arxiv.org/abs/2411.09595>
- [25] Yin F K, Chen X, Zhang C, *et al.* ShapeGPT: 3D shape generation with a unified multi-modal language model[J]. IEEE Transactions on Multimedia, 2025, 27: 4107-4120
- [26] Dong R P, Han C R, Peng Y, *et al.* DreamLLM: synergistic multimodal comprehension and creation[C] //Proceedings of the 12th International Conference on Learning Representations. Amherst: OpenReview, 2024: 1-37
- [27] Wu S F, Khasahmadi A H, Katz M, *et al.* CadVLM: bridging language and vision in the generation of parametric CAD sketches[C] //Proceedings of the 18th European Conference on Computer Vision. Heidelberg: Springer, 2024: 368-384
- [28] Grattafiori A, Dubey A, Jauhri A, *et al.* The Llama 3 Herd of Models[OL]. [2025-09-10]. <https://arxiv.org/abs/2407.21783>
- [29] Yang A, Li A, Yang B, *et al.* Qwen3 technical report[OL]. [2025-09-10]. <https://doi.org/10.48550/arXiv.2505.09388>