華東理工大學学报 自然科学版 Natural Science Edition
JOURNAL OF EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY

面向虚拟场景的大语言模型文本驱动连续地形生成方法研究

王唯佳 陆元军 吉斌 李建华 郭卫斌

A Text-Driven Continuous Terrain Generation Method for Virtual Scenes Based on Large Language Models

WANG Weijia, LU Yuanjun, JI Bin, LI Jianhua, GUO Weibin

在线阅读 View online: http://doi.org/10.14135/j.cnki.1006-3080.20251221001

您可能感兴趣的其他文章

Articles you may be interested in

基于TF-IDF和多头注意力Transformer模型的文本情感分析

Text Emotion Analysis Based on TF-IDF and Multihead Attention Transformer Model

华东理工大学学报（自然科学版）. 2024, 50(1): 129-136 http://doi.org/10.14135/j.cnki.1006-3080.20221218002

一种基于StyleGAN生成器的自监督人脸正面化模型

A Self-Supervised Face Frontalization Model Based on StyleGAN Generator

华东理工大学学报（自然科学版）. 2024, 50(6): 888-897 http://doi.org/10.14135/j.cnki.1006-3080.20231221001

基于二阶段对比学习的中文自动文本摘要方法研究

Automatic Chinese Text Summarization Based on Two-Stage Contrastive Learning

华东理工大学学报（自然科学版）. 2024, 50(4): 586-593 http://doi.org/10.14135/j.cnki.1006-3080.20230713001

基于BERT预训练模型的事故案例文本分类方法

Text Classification Method of Accident Cases Based on BERT Pre-Training Model

华东理工大学学报（自然科学版）. 2023, 49(4): 576-582 http://doi.org/10.14135/j.cnki.1006-3080.20220223002

基于音频内容和歌词文本相似度融合的翻唱歌曲识别模型

Cover Song Identification Model Based on Similarity Fusion of Audio Content and Lyrics Text

华东理工大学学报（自然科学版）. 2021, 47(1): 74-80 http://doi.org/10.14135/j.cnki.1006-3080.20191029002

基于图像描述的实验室气瓶危险场景辨识方法

Identification Method of Cylinder in Laboratory Dangerous Scene Based on Image Caption

华东理工大学学报（自然科学版）. 2023, 49(3): 410-418 http://doi.org/10.14135/j.cnki.1006-3080.20220124002

扫码关注公众号，获取更多信息！

文章编号：1006-3080(2026)04-0561-13 DOI: 10.14135/j.cnki.1006-3080.20251221001

面向虚拟场景的大语言模型文本驱动连续地形生成方法研究

王唯佳[1]， 陆元军[2]， 吉 斌[1]， 李建华[1]， 郭卫斌[1]
（1. 华东理工大学信息科学与工程学院，上海 200237； 2. 印孚瑟斯技术(中国)有限公司杭州分公司，杭州 310056）

摘要：随着游戏开发、虚拟现实(Virtual Reality, VR)及数字孪生等三维技术的快速发展，对高质量、大范围虚拟连续地形场景的自动化生成需求日益迫切。传统的手工交互方式操作繁琐且专业壁垒高，程序化生成方法虽能提升效率，但仍依赖图形界面和专业参数的调整，因而难以灵活捕捉用户语义意图。本文提出一种基于大语言模型(Large Language Model, LLM)的自动化文本驱动虚拟连续地形生成方法，旨在通过自然语言直接驱动生成过程，以实现高效、连续的虚拟地形场景构建。该方法采用多阶段任务分解策略，将地形生成流程拆分成7个步骤，融合提示词工程与程序化生成技术，构建从文本到大范围三维连续地形的端到端映射。针对LLM在复杂任务中的多跳推理缺陷、输出随机性及记忆丢失等局限，设计了跨阶段关键信息传递、错误重试与历史日志学习等机制，以提升系统可靠性和一致性。基于Unity引擎开发的原型系统验证了该方法的有效性，在4个典型地形主题上的测试结果表明，完整系统(Full Step System，FSS)的平均错误率仅为12.0%，相较于移除关键机制的对比系统，错误率显著降低了约19~47个百分点。在效率方面，生成单个预览地形区块(1 024×1 024顶点网格)的平均耗时约为42.14 s，资源消耗可控。对比实验进一步表明本文方法生成的虚拟地形在生成效率、视觉一致性与可扩展性方面均达到预期要求。该方法显著提升了生成效率与易用性，为非专业用户通过自然语言快速构建高质量、大范围三维地形提供了可行的解决方案。

关键词：人机交互；文本驱动；大语言模型；虚拟地形生成；程序化内容生成

中图分类号：TP18 **文献标志码：**A

虚拟现实(Virtual Reality, VR)、游戏开发及数字孪生等应用是虚拟空间的基础载体，虚拟地形生成的质量与效率直接决定了场景的真实感与用户体验。因此，面向VR、游戏等应用的高质量、大范围虚拟连续地形的自动化生成技术，成为推动相关领域发展的关键[1]。然而，传统的虚拟地形生成方法在效率与易用性方面面临显著挑战，探索能够快速生成高质量、大范围虚拟地形的新范式具有重要的研究意义。

传统的虚拟地形生成方法主要依赖于手工交互或程序化算法[2]，前者操作繁琐且专业壁垒高；后者从分形噪声[3]、物理模拟[4-5]演进到生成对抗网络(Generative Adversarial Network, GAN)技术[6-8]，虽然减少了参数依赖，提升了效率，但仍严重依赖专业参数调整或草图、高度场等结构化输入，未能从根本上降低交互复杂性，也难以灵活捕捉和实现用户的语义意图。因此，探索通过自然语言直接驱动虚拟地形生成的新方法，成为降低操作门槛、提升创作自由度的关键研究方向。

近年来，文本驱动生成研究为上述问题提供了

收稿日期：2025-12-21
基金项目：国家自然科学基金(62272164)
作者简介：王唯佳(2003—)，男，安徽人，硕士生，主要研究方向为计算机图形学。E-mail: 2630388173@qq.com
通信联系人：李建华，E-mail: jhli@ecust.edu.cn；郭卫斌，E-mail: gweibin@ecust.edu.cn
引用本文：王唯佳，陆元军，吉 斌，等. 面向虚拟场景的大语言模型文本驱动连续地形生成方法研究 [J]. 华东理工大学学报(自然科学版)，2026，52(4)：561-573.
Citation: WANG Weijia, LU Yuanjun, JI Bin, *et al*. A Text-Driven Continuous Terrain Generation Method for Virtual Scenes Based on Large Language Models[J]. Journal of East China University of Science and Technology, 2026, 52(4): 561-573.

新思路,现有方法可归纳为两类。第1类基于生成式模型(如扩散模型),典型代表如MESA框架[9],其利用全球遥感数据训练模型,能生成具有高真实感的2.5维地形表征(光学图像与高程图)。然而,此类方法输出为非结构化图像,需复杂的后处理方能转化为三维引擎可用资产;方法迭代式生成过程耗时较长,难以满足实时交互需求;生成思路偏向离散化构建,缺乏连续地形考量,且生成范围受限于固定输出尺寸,无法支持无限地形扩展。

第2类方法基于大语言模型(Large Language Model, LLM)的语义解析与程序化生成。该类方法通过LLM理解自然语言中的空间语义,并将其转化为程序化生成算法的参数或指令,进而驱动虚拟地形生成流程。例如,Text2VRScene系统[1]通过多阶段提示词工程与预定义提示词模板等机制,实现VR场景的端到端生成;Zhang等[10]采用思维链行为树,引导LLM参数化高斯-沃罗诺伊图以生成高度场;Liu等[11]则将LLM与遗传算法相结合,优化离散网格上的景观布局。

尽管第2类方法充分发挥了LLM在自然语言理解与任务编排上的优势,但现有研究普遍存在局限:Yin等[1]集中于单地块离散对象的空间拓扑构建,未解决大规模连续地形的几何生成与纹理映射问题;Zhang等[10]仅能生成灰度-高度图,缺乏纹理映射支持;Liu等[11]输出的宏观布局蓝图,依赖离散网格表达,难以实现连续自然过渡的地形细节。

综上所述,现有文本驱动方法在输出模态的完整性、地形生成的连续性以及大规模场景的可扩展性方面均存在不足,无法满足开放世界场景对无缝、沉浸式地形的高要求,亟须一种能够兼顾语义理解与工程可行性的端到端生成方案。

针对上述局限,本文延续并深化了第2类方法的思路,提出一种基于LLM与程序化虚拟连续地形生成管线深度融合的自动化文本驱动虚拟地形生成方法。该方法旨在克服现有LLM驱动方案在连续性与扩展性上的缺陷,实现从自然语言到可直接使用的大范围连续三维虚拟地形资产的端到端映射。本文的主要贡献如下:

(1)提出一个可扩展的语义驱动连续地形生成方法,通过多阶段任务分解与提示词工程,将文本描述映射为大范围虚拟连续地形,有效解决了现有方法在无限地形扩展与视觉一致性方面的不足。

(2)设计了历史日志学习、错误重试与跨阶段关键信息传递等机制,系统性地缓解了LLM在复杂生成任务中的多跳推理缺陷、输出随机性及记忆丢失等固有局限,提升了系统的可靠性。

(3)基于Unity引擎开发了原型系统,并通过严格的对比实验与性能评估,验证了该方法在生成质量、效率与稳定性方面的优势。

1 基于LLM的虚拟地形生成方法设计

本文的研究方法遵循“问题分析、方法设计、系统实现、有效性验证”的逻辑路径。首先通过总结过往文献,分析整理LLM在执行自动化生成任务中的局限与限制,进一步厘清LLM在大跨度地形生成任务中潜在的问题,进而考虑针对这些潜在局限的处理方法、设计方法、实现系统,以确保LLM在执行大规模虚拟连续地形生成任务上的可靠性。

1.1 LLM局限性分析

过往相关研究[12-13]对LLM执行生成任务的能力提出了一定的质疑。此外,Bang等[14]指出,LLM在缺乏外部知识库的情况下,会从其参数记忆中产生更多的外在幻觉;Borji[15]归纳了LLM在推理、事实错误、数学、编码和偏见等方面11种类型的失败;Yin等[1]分析LLM在进行VR场景生成时有着多跳推理缺陷、输出随机性等问题;Liu等[16]提出传统的LLM缺乏对生成结果的验证机制;Envgen框架研究[17]指出,直接使用LLM需数千次调用,并需通过模型蒸馏或轻量化设计平衡生成质量与实时性需求。综上,LLM存在以下几个固有局限:(1)LLM在多跳推理能力上存在缺陷,即在多步逻辑推理的复杂任务中表现欠佳。(2)LLM概率性生成机制导致相同输入可能产生格式或语义不一致的输出。(3)高级提示词工程技术依赖大量API调用与外部资源,复杂的提示词工程会导致API调用次数激增,因而调用成本问题显著。(4)在领域知识与数据依赖方面,LLM的上下文理解高度依赖训练数据的统计模式,若特定领域训练数据不足,易导致生成结果失真。(5)记忆丢失问题在长周期对话中尤为突出,LLM难以维持上下文一致性,易引发语义漂移。

1.2 方法设计

针对LLM在大范围连续地形生成流程中可能存在的局限性,本文提出一种基于提示词工程技术的生成方法,核心思想是将复杂任务拆解为连续可控可预览的子阶段任务链,并通过信息传递提示词工程技术等方式维持生成过程的连贯性,从而达到生成地形场景的目的,本文基于此方法在Unity中实现了对应系统。

Text2VRScene系统[1]在多阶段拆解与提示词工

程方面的开创性工作为本文提供了基础参考。然而, 该系统也存在应用领域从室内离散场景到室外连续地形的迁移, 以及 LLM 在执行生成任务中固有局限的问题。同时, Borne 等[9]、Zhang 等[10] 和 Liu 等[11] 的研究在大范围虚拟连续地形生成方面的短板更让其他研究者意识到, 区域性离散设计理念并不适用于大范围连续地形与动态加载的生成任务, 这面临一系列新挑战, 包括生成结果的连续性、对地理学规律的强约束以及实时渲染的性能需求等, 因此本文进行了以下几种针对性的设计与实现。

(1)为支持大范围连续地形的生成需求, 研究者使用了一套连续地形生成管线, 该管线通过算法优化与工程实现, 确保满足地形在几何连续性、视觉一致性和实时渲染性能等方面的要求。高程数据采用分形布朗运动(Fractal Brownian Motion, FBM)算法混合柏林噪声, 利用其连续性与自相似性特征, 构建自然过渡的地形高度场, 为地形连续性奠定基础; 在网格生成阶段引入边界缝合模块, 通过顶点位置共享与法线平滑插值技术, 消除相邻区块间的几何接缝。同时, 为保持边界区域细节的一致性, 边缘区域强制使用最高细节层次(Levels of Detail, LOD)的网格, 有效避免因 LOD 切换导致的视觉裂缝; 此外, 为支持无限地形的动态扩展, 系统采用多线程异步加载机制。高度图计算与网格构建任务分配至后台线程执行, 主线程仅负责视野范围内的区块渲染, 并通过动态卸载与加载策略, 实现大规模地形的流畅交互体验。

(2)针对 LLM 在复杂任务中的多跳推理缺陷问题, 本文借鉴 Text2VRScene 系统的多阶段任务分解策略[1], 将地形生成流程拆分为 7 个语义连贯的子阶段, 即语义扩充、路径生成、高度/噪声资产生成、纹理资产生成、网格资产生成、效果预览、地形生成。每个需要接入 LLM 的子阶段独立设计对应的提示词模板与验证规则, 通过降低单步任务复杂度, 能够有效控制 LLM 的推理错误累积风险。任务步骤的划分同时支持生成过程的中断与实时预览, 为后期人工微调提供重要技术支持。

(3)在提示词工程优化方面, 为降低 LLM 输出的随机性问题, 以及实现轻量化设计平衡生成质量与实时性需求, 本文采用预定义提示词模板与动态参数绑定相结合的混合策略。每个需要接入 LLM 的子阶段对应一套结构化提示模板, 集成多项提示词工程技术, 包括角色指定、示例引导、任务约束明确、关键词强化、语言风格规范化及专业术语解析增强, 以提升模型对领域任务的理解与输出一致性。各阶段提示词模板采用的分层设计如图 1 所示, 首先设定角色与任务情境, 其次明确任务描述与约束条件, 接着引入用户输入或扩展主题, 随后嵌入跨阶段传递的关键信息, 最后附以示例及其解析, 共同引导模型生成符合规范格式的代码输出。该模板的具体设计逻辑将在总体流程与提示词设计部分中详细叙述。

任务：{预定义角色扮演与情景构筑，简要交代任务}

约束：{对任务要求的详细描述}

主题：{用户输入或经过扩充的主题}

关键信息：{当前任务相关的历史关键信息}... ...

示例：{样例示例与用户选择历史}

示例解析：{输出代码含义解析}... ...

图 1　各阶段提示词模板

Fig. 1　Prompt templates for each step

(4)为实现生成过程的可追溯性、调试优化与辅助 API 调用轻量化管理, 系统建立了完整的日志记录机制。所有 LLM 交互记录与错误信息均以结构化格式保存, 每条日志包含原始提示、模型响应及时间戳等关键信息。同时, 系统通过 Unity 编辑器扩展实现实时预览机制, 用户可在生成过程中实时查看地形效果, 并可通过停止按钮随时中断任务流和参数调整。这一设计有效平衡了自动化生成效率与人工控制需求, 为迭代优化提供了便利。

(5)为实现生成过程的持续优化与个性化适配, 针对 LLM 领域知识与数据依赖问题, 系统设计了基于日志的历史学习功能。用户可筛选结构化的历史日志, 并将其作为 LLM 的附加输入参考。此举旨在为 LLM 提供具体案例, 以弥补其在上下文理解上对训练数据的依赖, 并增强对用户个性化偏好的适应能力。结合信息传递机制, 该学习方式有助于缓解在连续对话中 LLM 对意图变化敏感度不足的问题, 从而稳定输出质量。

(6)研究者还设置了出错重试与处理机制, 当检测到响应不符合预设规范或运行中出现的各种错误(如 API 连接中断)时, 将自动尝试重复当前步骤。该机制通过输出验证与修复错误, 可以有效降低因单点故障导致的流程崩溃风险, 提升系统鲁棒性。

(7)Yin 等[1] 针对 LLM 上下文记忆能力不足的问题提出了关键信息传递机制, 这在本文中表现为一种上下文缓存机制, 通过从历史对话中提取地形类型、噪声参数范围、纹理映射规则等核心信息, 并将其编码为结构化摘要嵌入后续提示词。例如, 地形网格生成阶段的提示词将包含历史步骤的关键参

数的引用。通过强制 LLM 关注历史约束，显著降低语义漂移概率，确保多步骤任务的一致性。

1.3 总体流程与提示词设计

1.3.1 总体流程概述 虚拟地形生成的总流程如图 2 所示。首先，流程始于自然语言输入与语义扩充。在此阶段，LLM 首要任务是对用户可能较为简略的输入(如“崇山峻岭”)进行解析和丰富。通过结构化的提示词模板，LLM 被引导从地形类型、高度特征、色调纹理、独特元素、环境要素及功能性等多个维度对主题进行扩写，旨在生成一份详尽、结构化的地形描述，为后续所有步骤提供准确的语义基础。

在获得丰富的主题描述后随即进入工程配置阶段。此阶段主要完成项目唯一标识符的生成与标准化的目录结构创建，以确保每次生成任务资源的独立性与可管理性。为实现该流程的自动化控制，提示词工程采用多重约束设计范式，给出示例模板并严格限定项目标识输出格式，剔除额外字符，从而确保大语言模型输出的标题能准确反映扩展主题的核心特征。

随后，流程步入核心的资产生成环节，该环节依次包含高度/噪声资产、纹理资产和网格资产的生成。这 3 个阶段具有相似的逻辑模式：LLM 均依据前序步骤传递的信息(主要是扩展后的主题描述以及已生成资产的关键参数)，结合针对性的提示词模板输出标准化配置下的资产代码。这些资产代码随后被响应解析器提取，并实例化为具体的工程资产文件。值得注意的是，每个阶段产出的关键参数(如噪声参数、纹理层配置)都会通过序列化接口暴露，并作为“关键信息”嵌入下一阶段的提示词中，以确保多步骤间参数的一致性与关联性。

最后，在地形预览与生成阶段，地形生成器会调用前述步骤创建的所有资产代码文件，整合高度图生成、纹理混合、网格构建与 LOD 优化等模块，调用 Unity 引擎的实时渲染功能，生成最终的可交互、可无限扩展的连续地形预览。

1.3.2 关键步骤的提示词设计策略 为便于更直观地理解，本节将深入剖析两个核心阶段：语义扩写，高度/噪声资产生成阶段的提示词设计策略。

语义扩写阶段是整个流程的起点，其设计质量直接决定了后续所有步骤的语义基础是否牢固。该阶段的核心挑战在于：用户的初始输入(如“崇山峻岭”)往往语义密度不足且充满模糊性，若直接交由 LLM 自由发挥，极易因“输出随机性”和“领域知识依赖”问题而产生偏离常识或难以程序化的描述。图 3 所示的提示词模板，正是为约束此种随机性并注入该领域的初步知识而构建。

语义扩写阶段的设计策略目标是生成一份结构清晰、要素完备、可供后续程序化流程精确解析的“技术规格说明书”，具体而言，该模板通过多重设计来确保输出质量。首先，通过角色指定(“假设你是一个地形设计专家”)为 LLM 设定专业语境，引导其以领域专家的思维模式进行思考。其次，结构化约束则明确要求输出必须包含“地形的类型、形状和主要特征”、“主要高度特征”、“主要色调和纹理特征”等 6 大核心维度，这实质上是将地形描述的抽象任务进行了原子化分解，以应对 LLM 的多跳推理缺陷，确

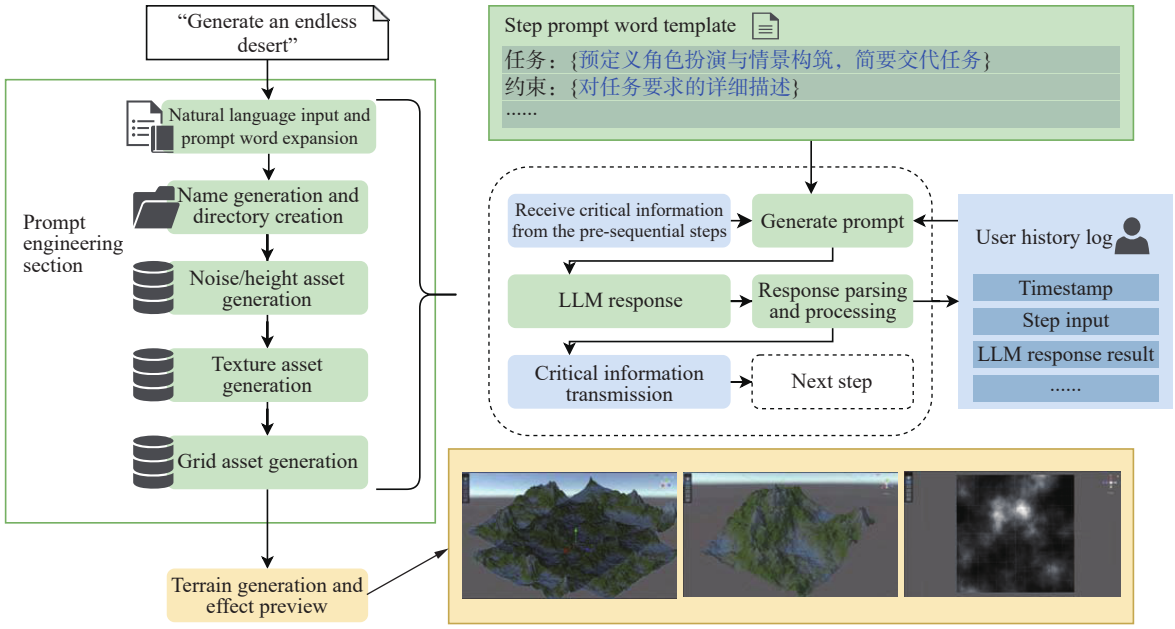


图 2 虚拟地形生成总流程示意图

Fig. 2 Schematic diagram of the overall virtual terrain generation process

保生成描述的完整性与结构性。最后，示例引导为用户输入“崇山峻岭”提供了标准化的输出范例，如“高耸的山脉、陡峭的悬崖、深邃的峡谷”等，此举旨在为 LLM 建立正确的输出范式，显著降低其因输出随机性而导致的描述失真风险。

高度/噪声资产生成阶段则面临着不同的挑战。此阶段需要将上一步得到的语义描述，准确地转化为一系列可控制地形形态的程序化参数（如噪声缩放比例、八度数量等）。其核心难点在于克服 LLM 的“领域知识依赖”，即 LLM 本身并不理解“噪声缩放比例”与“崇山峻岭”的地貌特征之间应建立何种数值映射关系。为此，将图 4 所示的提示词模板设计成旨在系统性地引导 LLM 完成从语义到代码的可靠转换。

高度/噪声资产生成阶段模板的设计紧密围绕提示词工程优化原则展开。首先，通过角色指定为 LLM 设定明确的专业语境。其次，任务约束被具体化为两项核心指令：“生成对应的噪声/高度资产代码”和“给出其他在生成符合主题描述的地形时会被用到的变量”。这两项约束并非空泛的要求，而是通过提供一个完整且符合引擎资产序列化格式的代码模板来实现。在该模板中，所有需要 LLM 填充的参数（如 scale, octaves 等）均已明确定义其名称、含义、取值范围及示例。这种设计实质上是将 LLM 的类比推理与代码生成能力相结合，无需从零开始“理解”地形学知识，而是依据 LLM 强大的代码补全和模式识别能力，参考模板中预设的参数定义和示例，将输入的语义描述类比映射到合适的数值区间。

同时，该阶段也是关键信息传递机制的重要应用节点。模板中相关的参数描述将会转化为“你可以结合的相关高度变量含义以及变量值如下”的部分，并嵌入后续步骤提示词模板中。此设计确保了在多轮生成或复杂任务中，能够有效应对 LLM 的“记忆丢失”问题，维护上下文一致性。

2　实验设置与结果分析

基于前文所述方法，本文开发了基于 LLM 的文本驱动连续地形生成系统，该系统可以通过输入简单的文字提示词生成对应的连续地形。

此次系统实现使用了 Unity 2022.3.48引擎，以 C# 语言为主要编程语言。配合 Unity 引擎的 Scriptable Object 数据持久化技术，系统将地形生成参数配置模板化并封装为 Scriptable Object 资产，支持多套配置的快速切换与历史方案的回溯复用，避免重复计算。修改 Scriptable Object 参数可实时更新地形，便于调试验证。系统使用 C#编写协程请求脚本，接入 GPT-4o、DeepSeek 等 LLM API，并通过协程迭代器实现流程的分帧加载。

2.1　实验设置

2.1.1　评价指标　为了验证系统是否能够顺利执行自动地形生成任务，在评估指标方面，本文针对 LLM 的随机性采用了错误率（Error Rate）来衡量自动

任务：假设你是一个地形设计专家，需要为一个地形生成系统提供详细的输入描述，你必须严格遵守以下约束。

约束：1. 输出的地形描述必须基于给定主题，细节丰富，并包含以下方面的内容：

- 地形的类型、形状和主要特征（如山脉等）。
- 地形的主要高度特征（如高耸的山脉等）。
- 主要色调和纹理特征（如绿色，布满植被等）。
- 地形的独特元素（如特殊植被、岩石分布等）。
- 适合的气候条件、光照和风貌等环境要素。
- 功能性描述（不要出现人工建筑等）

2. 描述应结构化，便于程序将描述转化为代码逻辑。

3. 扩充给定的地形描述，使其更符合主题，并尽可能提供细致的场景信息……(更多约束)

主题：{用户输入}……

示例：{样例示例与用户选择历史}

示例解析：{示例要点分析}……

图 3　自然语言输入与提示词扩写步骤的提示词模板

Fig. 3　Prompt template for natural language input and prompt expansion step

任务：假设你是一名地形设计专家，需要为一个地形生成系统提供详细的高度噪音参数输入，你必须严格遵守以下约束。

约束：1. 没有用户帮助。

2. 你应该输出依据于给定主题的相关地形的噪声高度资产代码

3. 你应该给出其他在生成符合主题描述的地形时会被用到的变量，请充分考虑给定的主题以及各个变量的含义与作用，生成对应的变量的值，相关的约束将会在之后给出

4. 你应该只输出如下代码，不要添加除此之外的任何额外解释，确保响应可以由Unity解析

此次给出的主题为：{步骤一中扩充的主题}。

{资产代码格式模板}……

参数解释(节选核心部分展示)：

- scale：噪声缩放指数，最小为0.01f (e.g., 50)
- octaves：噪声八度音阶的数量，最小为1(e.g., 6)
- persistence：噪声持久性值，在[0,1]之间(e.g., 0.5)
- lacunarity：噪声间隙度值，最小为1(e.g., 2)
- heightMultiplier：网格高度乘数，生成网格时在范围为[0,1]的高度基础上乘上这个值即为网格的最终高度(e.g., 50)……

示例：{样例示例与用户选择历史}

示例解析：{含义解析}……

图4 高度/噪声资产生成的提示词模板

Fig. 4 Prompt template for height/noise asset generation

化生成系统的可靠性，该选择基于以下方面考虑：

首先，本文任务特性与Text2VRScene高度相似，均属于LLM驱动的多阶段生成系统。如Yin等[1]所述，此类系统的核心挑战是LLM在复杂任务中的多跳推理缺陷、输出随机性及记忆丢失等问题。本文系统与Text2VRScene均采用多阶段任务分解策略，将生成流程拆分为语义连贯的子任务，且均面临资产代码生成、主题一致性维护等共性挑战。因此，Text2VRScene所采用的错误率指标能够直接适用于本文场景，能有效衡量LLM在连续地形生成任务中的稳定性。

其次，连续地形生成的输出特性限制了传统客观指标的适用性。与传统地形生成方法(如基于GAN或物理模拟的方法)不同，本文的输出是程序化生成的、无限扩展的三维地形序列，而非固定尺寸的高度图或纹理贴图。此类输出的“正确性”难以通过像素级相似度(如峰值信噪比(Peak Signal-to-Noise Ratio, PSRN)、结构相似性指数(Structural Similarity Index Measure, SSIM))或几何误差(如Chamfer Distance)等传统指标量化，因为其本质是动态生成过程而非静态资产。相反，用户更关注系统能否可靠地将语义意图转化为可用的地形资产。因此，错误率这一指标更能反映系统的实用价值，它直接衡量了从用户输入到最终地形的端到端生成的成功率。

错误率的分类定义需紧密结合虚拟地形生成的工程实践，本文进一步将错误定义并分类为以下几种情况：系统无法正常运行获得结果(系统运行错误)；生成的地形资产文件无法被Unity解析(资产代码生成错误)；生成的地形偏离用户给出的提示词主题且场景无法被用户理解识别(主题偏离错误)。以上情况需排除网络或者其他因素引起的系统崩溃问题，并定义错误率为系统在执行任务时遇到的概率。

该指标能够有效反映LLM在复杂任务中的稳定性，且更符合本文动态生成、语义驱动的任务特性，已被同类工作验证为适用于生成系统的可靠性度量[1]。此外，实验还将额外记录单次生成耗时与资源消耗(CPU、内存)，作为性能指标数据以评估系统效率。

2.1.2 *实验方法* 为全面验证所提方法的有效性，实验设计遵循系统性原则，分为消融实验、性能实验与对比实验这3个部分，旨在从可靠性、效率及先进性等多个维度评估本系统的综合性能。

消融实验是本次评估的核心，其目的在于定量分析本文所提各项关键机制对系统整体性能的贡献度。为此，在完整系统(Full Step System, FSS)的基础上，通过控制变量法构建了4个对比系统。FSS作为基准，整合了本文设计的全部关键机制，包括错误重试与处理、基于日志的历史学习、跨阶段关键信息传递以及结构化的提示词工程优化等技术。

与之相对，无错误处理与重试(No Error

Handling and Retry, NEHR)系统移除了错误检测与自动重试机制, 旨在验证该机制对减弱 LLM 输出的随机性、提升流程鲁棒性的作用; 当步骤执行过程中出现输出格式异常或运行错误时, NEHR 系统将直接终止流程, 而不会尝试修复或重新执行。无样例与历史学习(No Sample prompts and Historical learning, NSH)系统则移除了提示词模板中的示例引导功能, 并禁用了用户历史日志的加载与学习, 用以评估示例引导与个性化适配对生成质量稳定性的影响; 无关键信息传递(No Critical Messaging, NCM)系统的核心改动在于切断了跨阶段的关键参数(如地形类型、噪声参数、纹理配置等)传递链路, 以检验该机制在维持多步骤任务语义一致性与逻辑连贯性方面的必要性; 无提示词工程优化(No Prompt Engineering Optimization, NPEO)系统移除了所有高级提示词工程技术(如角色指定、任务约束明确、示例解析等), 仅保留最基本的输出格式要求, 用以探究提示词工程在引导 LLM 理解领域任务、降低输出随机性方面的基础性价值; 最后, 无阶段分割(No Stage Segmentation, NSS)系统没有将生成任务拆分成多个步骤, 而是直接一次性要求 LLM 生成全部资产代码, 由于缺乏阶段分割, LLM 单次推理负担过重, 导致各个环节都出现了输出解析错误积累的情况, 系统在绝大多数尝试中均无法正常运行, 这从反面印证了阶段分割机制的重要性。

在具体执行方面, 将针对上述 5 个系统(FSS, NEHR, NSH, NCM, NPEO), 在 4 个典型地形主题(主题 1“崇山峻岭”、主题 2“绵延荒漠”、主题 3“冰川极地”、主题 4“零星海岛”)上分别重复执行 50 次生成任务, 并详细统计各系统的错误率, 从而精确量化每个被移除机制对系统可靠性的影响程度。生成效果展示中额外补充了 ChatGPT 网页系统, 此系统直接使用 ChatGPT 网站, 改写提示模板指导用户简单使用 Unity 的 Terrain 系统。此系统需人工介入处理, 将自动化系统与其比较意义不大, 这里主要用该系统展示 LLM 在地形生成任务方面的潜力。

性能实验侧重于评估系统的效率与资源消耗。在 FSS 系统完成上述可靠性测试的基础上, 将同步记录其生成单个预览地形区块(1 024×1 024 顶点网格)的总耗时、各子阶段(如语义扩写、资产生成等)的时间占比、CPU 峰值占用率以及内存增量。这些数据将直观地展现系统在实时性(响应速度)与资源效率方面的表现, 为评估其在实际应用中的可行性提供关键依据。

对比实验则旨在将本文方法与现有代表性工作进行横向比较, 以凸显其独特优势与贡献。本研究选取了同属 LLM 驱动生成范式的 Text2VRScene 系统[1] 作为主要对比对象。通过输入相同的主题提示, 将从生成地形的基本特性(如连续性、可扩展性)、视觉效果一致性以及生成效率等多个层面进行综合对比, 从而在更广阔的视野下验证本文方法在解决虚拟连续地形生成这一特定任务上的先进性与实用性。

2.2　消融实验结果分析

为量化评估本文所提各项关键机制的有效性, 首先对消融实验的整体结果进行了统计分析。表 1 示出了 FSS 与对比系统(NEHR、NCM、NPEO、NSH)在 4 个标准主题下的错误率对比结果。FSS 系统取得了最低的平均错误率(12.0%), 相较于表现最差的 NPEO 系统(59.0%), 错误率降低了 47 个百分点。即使在随机主题测试中(表 2), FSS 的错误率(14.0%)也显著低于对比系统(26.0%~62.0%)。这些结果初步验证了 FSS 系统在可靠性方面的显著优势。

然而, 整体错误率尚不足以揭示各机制如何针对性地改善系统的不同方面。为此, 本文进一步统计了各系统的错误类型分布(表 3), 并绘制了堆积直方图(图 5)。如表 3 和图 5 所示, 不同对比系统的错误类型呈现出截然不同的分布模式, 这为了解特定机制失效所带来的具体影响提供了关键线索。下文将结合表 3 数据, 对每个对比系统进行深入剖析, 以阐明错误重试、历史学习、关键信息传递及提示词工程等机制在提升系统鲁棒性与生成质量中的作用。

2.2.1　NEHR 系统　如表 1 所示, NEHR 系统在 4 个主题下的平均错误率为 41.5%, 显著高于 FSS 的 12.0%。如表 3 和图 5 所示, 资产代码错误在此系统中占比最高(48.2%)。经分析, 此类错误主要源于生成的资产代码包含冗余的 Markdown标记等格式问题, 导致系统无法正确解析。此外, 少量运行错误(1.2%)可归因于移除了文件路径合法性检查机制, 致使 LLM 输出无法进行有效约束。这表明错误处理机制通过校验输出格式、路径规范与错误重试, 有效降低了因 LLM 输出随机性引发的资产代码解析错误和系统运行异常问题。

2.2.2　NCM 系统　在移除关键信息传递机制的 NCM 系统中, 偏离主题错误占比高达 96.8%(表 3), 且其平均错误率(30.8%)较 FSS(12.0%)高出 18.8 个百分点, 生成内容的质量与准确性显著下降。分析认为, 缺失跨步骤的核心参数(如噪声范围、纹理映射规则)传递, 导致 LLM 无法准确控制资产变量间的关联性, 难以维持地形生成的视觉一致性与逻辑和谐性。同时, NCM 系统中出现的少量运行错误

表1 不同系统在各个主题下的错误率比较

Table 1 Error rate comparison on different topics of different systems

System	Error rate/%				
	Topic 1	Topic 2	Topic 3	Topic 4	Average
NEHR	52.0	38.0	40.0	36.0	41.5
NCM	33.0	34.0	34.0	22.0	30.8
NPEO	74.0	64.0	52.0	46.0	59.0
NSH	44.0	58.0	54.0	58.0	53.5
FSS	**8.0**	**16.0**	**14.0**	**10.0**	**12.0**

表2 不同系统在随机主题测试中的错误率

Table 2 Error rates on random topics of different systems

System	Error rate/%
NEHR	56.0
NCM	26.0
NPEO	62.0
NSH	56.0
FSS	**14.0**

表3 不同系统的错误类型分布

Table 3 Error type distribution of different systems

System	Error rate/%		
	Runtime	Code	Off-topic
NEHR	1.2	48.2	50.6
NCM	1.6	1.6	96.8
NPEO	36.4	6.8	56.8
NSH	0.9	47.7	51.4
FSS	0	0	100.0

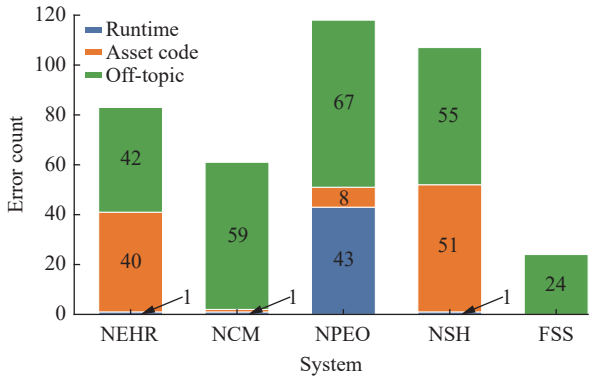


图5 不同系统的错误类型分布

Fig. 5 Error type distribution of different systems

(1.6%)和代码错误(1.6%)同样源于上下文关键信息缺失导致的输出异常问题。

2.2.3 NPEO系统 NPEO系统的各个步骤的提示词模板中不含有情景构筑角色扮演等无提示词工程约束技术,只规定了必需的输出格式。如表1所示,NPEO系统在所有对比系统中表现最差,说明提示词工程优化技术在有效发挥LLM任务执行自动化内容生成任务的潜力方面至关重要。

值得注意的是,在NPEO系统中,运行错误率大幅上升至36.4%(表3),平均错误率(59.0%)较FSS(12.01%)高出47个百分点。其主要原因在于缺乏角色扮演、详细任务描述及领域知识约束,导致LLM无法充分理解任务意图并输出有效内容。典型表现为文件路径生成步骤频繁产生过长响应,超出系统处理上限,引发后续文件读取失败和系统崩溃。

此外,偏离用户意图的错误仍占NPEO总错误数的56.8%,具体流程追踪显示,移除地理常识等专业约束后,LLM输出频繁违背基本逻辑,如在沙漠场景中选择大面积的植被纹理、在低海拔区域选择雪地纹理却在高海拔区域选择草地纹理等。这印证了提示词工程优化可以在有效发挥LLM的语言解析潜力与任务执行能力的同时,减少因为输出随机性、缺乏专业知识与数据依赖性带来的不确定性。

2.2.4 NSH系统 在移除样例提示与历史日志学习机制的NSH系统中,资产代码解析错误成为其主要错误类型,占比达47.7%(表3)。原因是直接提供脚本代码与参数替代样例学习,使LLM输出极不稳定,不能严格遵循指定格式(以复杂的纹理参数错误为甚),导致生成的资产代码无法被Unity引擎解析。尽管部分资产可经引擎修复后使用,但最终效果仍不理想。相比之下,在FSS中占比最大的偏离主题错误在NSH系统中降至51.4%。这表明历史学习机制通过提供结构化参考样例,能有效约束LLM输出随机性,提升系统稳定性。

为验证历史学习机制对减少离题错误的效果,本文针对FSS系统进行了补充实验:对4个主题各执行50次,每次实验均精心选择两个具有参考价值的历史日志和资产作为额外输入(标记为System With Selected Historical Logs, SWSHL),统计SWSHL系统的离题错误率,并与未使用历史日志的FSS基线以及无历史学习的NSH系统进行比较,离题错误率结果如表4所示。

由表4可知,引入SWSHL后,系统的平均离题错误率进一步降至4.5%,低于未使用历史日志的FSS系统的12.0%,远低于完全无历史日志学习机制的NSH系统的27.5%。这一结果表明本文提出的基于日志的历史学习机制对缓解离题错误具有显著效

表 4　基于日志的历史学习机制对离题错误率的影响

Table 4　Effect of historical learning mechanism based on logs on off-topic error rate

System	Error rate/%				
	Topic 1	Topic 2	Topic 3	Topic 4	Average
NSH	28.0	38.0	26.0	18.0	27.5
FSS	8.0	16.0	14.0	10.0	12.0
SWSHL	**6.0**	**6.0**	**4.0**	**2.0**	**4.5**

果。将用户筛选的过往日志作为上下文示例输入 LLM, 模型能够更准确地捕捉用户偏好、任务惯例以及领域特定约束(如地形纹理与高度的合理对应关系), 从而有效减少因语义漂移或常识性错误导致的生成结果偏离用户原始意图的情况。

SWSHL 系统相较于 FSS 系统的错误率降低, 证明了在 FSS 系统基础上, 历史学习机制提供了额外的、针对性的约束能力。它弥补了静态提示模板在覆盖用户个性化需求和复杂地形语义细微差别方面的不足, 辅助模型理解特定用户或特定地形类型的关键特征映射。反之, NSH 系统的高离题错误率(27.5%)从反面印证了历史/样例信息缺失对模型理解用户意图的负面影响。缺乏参考依据模型的输出具有更高的随机性和偏离主题的风险。

2.2.5 生成效果对比　图 6 示出了不同系统基于 4 个主题生成的地形预览效果。选取了 ChatGPT-4o 网页版、NPEO、NSH 系统和 NCM 系统作为 FSS 系统的对照。由于 NEHR 系统的错误类型大部分是系统运行崩溃, 且生成过程与本系统相同, 不具备展示意义, 故而省略。

由图 6(a) 可以看出, LLM 具备解析用户主题输入以及理解地形生成任务逻辑的能力, 可以设置与地形相关的高度、纹理等参数, 并将其组合成符合用户要求的结果。在图 6(b) 所示的 NPEO 系统中, 由于去除了提示词模板中的角色扮演、地理常识等约束, 可以明显地观察到 LLM 生成的结果在多处违背了常识, 因而不符合用户的预期。如图 6(c) 所示, NSH 系统输出极不稳定, 离题率与资产错误率均较高。由于 LLM 对于数据的高度依赖性, 要得到理想结果, 需要更进一步的手动调整。在 NCM 系统中, 由于缺乏步骤间的信息传递, 每个步骤都变得独立, 导致生成的地形缺乏整体和谐性。这在图 6(d) 主题 1(“崇山峻岭”)中表现明显: 水体纹理设置异常, 违背用户预期。整合了所有机制的 FSS 系统 (图 6(e)) 成功地理解了简单的地形主题逻辑并将其扩充为详细的描述, 与对应参数相关联, 并最终获得了令人满意的效果。

2.3　性能实验与分析

为全面评估 FSS 系统效率, 本文对 4 个典型主题(“崇山峻岭”、“绵延荒漠”、“冰川极地”、“零星海岛”)进行了 FSS 系统的性能测试。测试环境为: AMD Ryzen 9 5900HX with Radeon Graphics CPU, NVIDIA GeForce 3080 Laptop GPU, 16GB RAM, Unity 2022.3.48, GPT-4o API, 测试结果如表 5 所示。在时间效率方面, FSS 系统生成单个地形区块预览(1 024×1 024 顶点网格)平均耗时 42.14 s。其中“崇山峻岭”(主题 1)因其地形结构相对常规, 耗时最短(33.46 s); 而“零星海岛”(主题 4)需处理更复杂的高度变化与纹理搭配选择, 耗时最长(59.49 s)。

在资源消耗方面, CPU 峰值占用率稳定在 1.37%~1.44% 之间, 计算负载较低; 内存增量平均为11.93 MB, 主要源于 Scriptable Object 资产的序列化存储。不同主题间内存消耗(8.38~15.08 MB)差异与纹理复杂度呈正相关, 海岛和冰川主题因需要更多潜在的纹理层而内存消耗略高。

各个步骤的时间消耗与占比如图 7 所示。按 4 类主题不同阶段的平均耗时换算, 高度/噪声资产生成阶段与纹理资产生成阶段是系统运行中的主要耗时环节, 二者合计约占总耗时的 66.9%, 说明系统的时间开销主要集中在地形参数生成与视觉资产配置两个环节。其中, 高度/噪声资产生成平均耗时约为 14.2 s, 在“零星海岛”(主题 4)中达到 20.8 s。这是因为海岛地形通常同时包含陆地区域、水域边界、局部高度差变化以及多尺度噪声叠加等要素, 参数组合相对复杂, 因此带来了更高的生成开销。纹理资产生成平均耗时 14.0 s, 占比保持在约 32.0% ~ 34.3%, 整体波动较小, 表明不同主题虽然对应的纹理类型和视觉风格存在差异, 但纹理层起始高度、纹理选择、色调强度及层间过渡等参数生成过程相对稳定。语义扩写阶段平均占比为 15.8%±0.7%, 在“冰川极地”和“零星海岛”主题中略高(分别约为 16.5% 和 16.1%), 说明当地形主题包含更丰富的地貌语义或

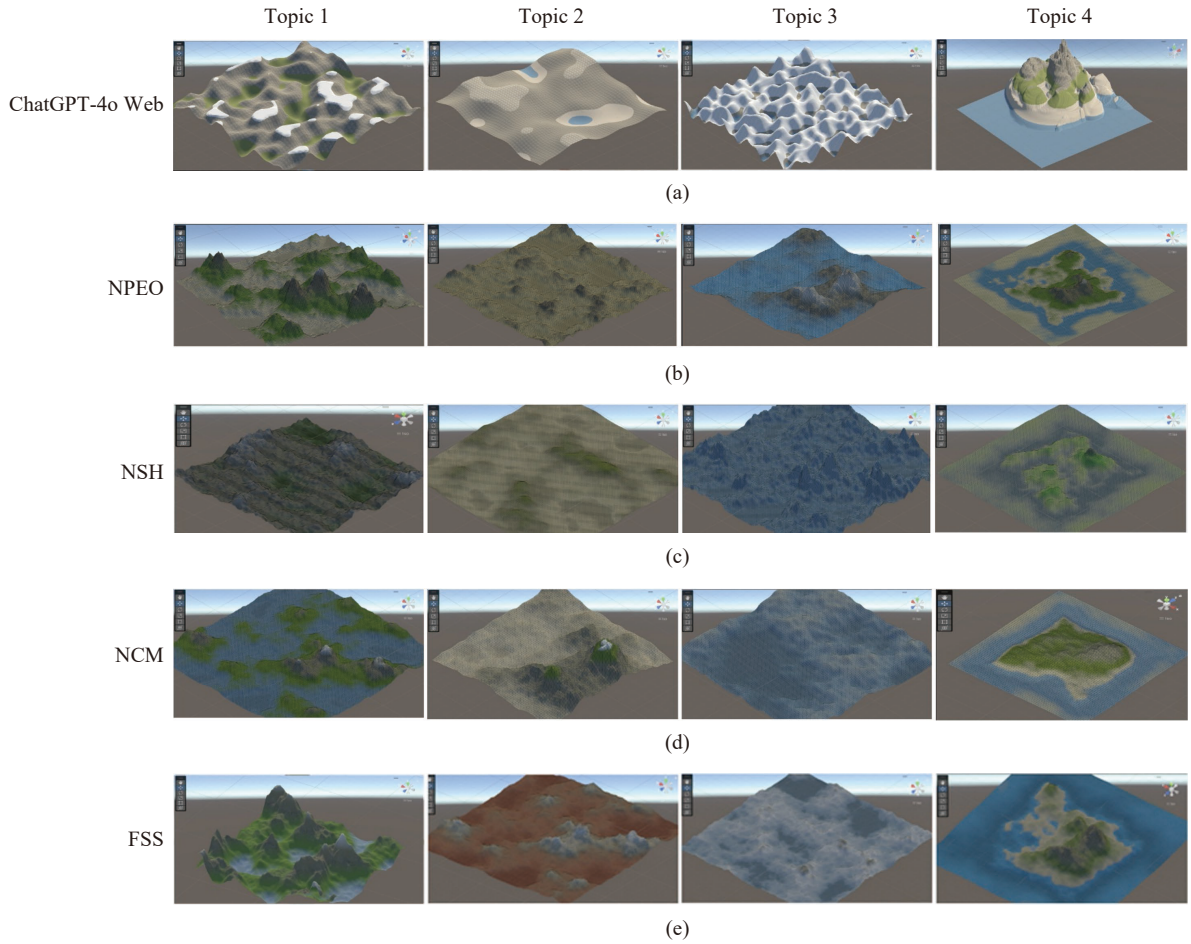


图 6 不同系统生成的地形预览效果

Fig. 6 Terrain preview effects generated by different systems

表 5 FSS 系统性能测试结果

Table 5 Performance test results of FSS system

Topic	Total time/s	CPU peak usage/%	Memory increment/MB
Topic 1	33.46	1.37	8.38
Topic 2	33.72	1.39	10.74
Topic 3	41.88	1.44	13.55
Topic 4	59.49	1.42	15.08
Average	42.14	1.41	11.93

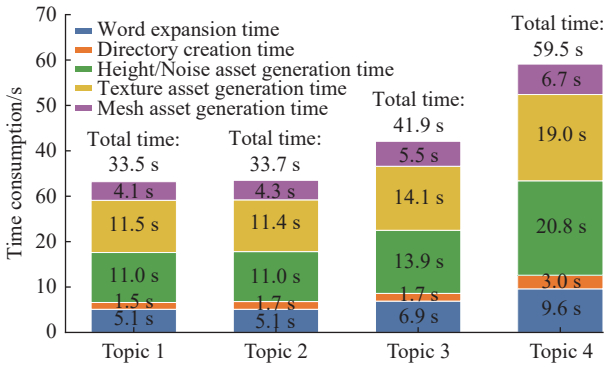


图 7 系统各个步骤耗时与占比

Fig. 7 Time consumption and proportion of each step in system

更复杂的空间关系时，LLM 需要生成和组织更多描述信息，处理时间也会相应增加。相比之下，网格资产生成阶段占比 12.3%±0.7%，路径生成阶段仅占 4.7%±0.5%，二者占比较低且波动不大，说明几何网格构建与文件路径组织流程整体可控。综合来看，系统主要性能开销集中在高度/噪声和纹理等核心地形资产生成环节，这也与连续地形生成任务中地貌结构和视觉表现共同决定生成复杂度的特点相一致。

综上，本文提出的方法在保证生成质量的前提下，可实现低于 45 s 的端到端生成时间，较传统手动流程（通常需约 10 min）显著提升。资源消耗维持在较低水平，展现出实时交互的潜力。

2.4 对比实验分析

为在更广阔的视野下评估本文方法的先进性与实用性，本节将 FSS 系统与同属 LLM 驱动生成范式的 Text2VRScene 系统[1] 进行横向对比。实验在相同的硬件环境与网络条件下，针对同一主题（“崇山峻岭”）分别执行了 50 次生成任务，并从生成效率与功能特性两个维度进行综合分析。

在生成效率方面，定量统计数据揭示了

Text2VRScen 和 FSS 两种方法在性能上的显著差异。如表 6 所示, FSS 系统生成单个地形区块的平均耗时为 22.99 s, 而 Text2VRScene 系统的平均耗时长达 423.31 s, 约为 FSS 系统的 18.4 倍。同时, 在计算资源消耗方面, FSS 单次任务的平均 Token 用量约为 9645, 显著低于 Text2VRScene 系统的 15381。

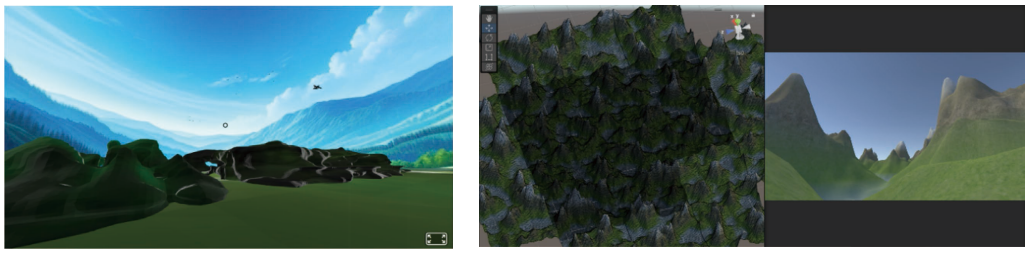
这一效率差距主要源于两系统生成逻辑的根本不同。Text2VRScene 依赖于 LLM 进行多轮复杂推理以生成静态场景布局描述, 并常常需要调用外部模型生成天空盒等资源, 其迭代过程导致了高昂开销。相较之下, 本文 FSS 方法通过精心设计的任务分解策略, 将复杂的地形生成任务拆解为一系列相对简单、可控的程序化资产生成步骤, 极大降低了 LLM 单步推理的复杂度, 从而实现了高效的流水线作业。

表 6　FSS 与 Text2VRScene 系统的性能对比 (同一主题, 50 次任务平均值)

Table 6　Performance comparison of FSS with Text2VRScene system (Same topic, average of 50 runs)

System	Average generation time/s	Average Token usage	Core output modality (modality)
Text2VRScene	423.31	15381	Discrete scene layout (Model objects + Skybox)
FSS	22.99	9645	Continuous & extensible terrain assets

更为关键的是二者在输出模态与功能特性上的根本区别。Text2VRScene 系统主要关注离散对象的空间拓扑关系建模与基础周期动画绑定(如旋转、位移等简单运动), 但缺乏对连续空间地形生成的技术支持。如图 8(a) 所示, Text2VRScene 场景由山体、飞鸟和地面等离散模型经缩放排列构成, 生成的是一个范围固定、不可扩展的场景“快照”, 该方法难以支持视点的连续移动与地形的无缝延伸。相比之下, 本文 FSS 系统通过设置地形生成管线分形噪声算法生成连续高度场, 结合动态 LOD 优化与异步区块加载技术, 实现了基于视点驱动的无限地形扩展(图 8(b) 左半部分示意)。生成效果如图 8(b) 右半部分所示, 虚拟地形具有连续的高度场和自然融合的纹理, 能够根据视点实时加载新区块, 实现了真正的无限地形扩展能力。这种从离散场景布局到连续地形生成的范式转变是本文的核心价值。



(a) Text2VRScene system　　(b) Our system

图 8　Text2VRScene 与本文 FSS 系统生成效果对比

Fig. 8　Comparison between the effects generated by Text2VRScene and FSS system in this paper

3　结束语

本文提出了一种基于 LLM 与程序化生成相结合的文本驱动地形生成方法。实验结果表明, 集成全部优化机制的完整系统 FSS 在 4 个典型地形主题测试中的平均错误率仅为 12.0%, 相较于移除关键机制的对比系统, 错误率降低了 19~47 个百分点。在生成效率方面, 系统生成单个预览地形区块(1024×1024 顶点网格)的平均耗时约为 42.14 s, 资源消耗可控, 在保证生成质量的同时展现了良好的实时交互潜力。与 Text2VRScene 系统的对比进一步表明, 本文方法不仅在生成效率上具有数量级优势, 更实现了从离散场景布局到连续地形生成的范式转变, 凸显了其在解决特定任务上的独特价值。

尽管本文方法取得了令人满意的成果, 但仍存在一定的局限性。当前系统主要侧重于宏观地貌与基础纹理的生成, 对植被、岩石、水体等生态细节以及人造地物的建模能力尚有不足, 影响了场景的丰富度与真实感。同时, 地形生成的算法核心仍主要基于 Perlin 噪声及其变体, 其所能表达的地形特征谱系相对有限。

未来的研究工作将重点从以下几个方向展开: 首先, 将探索集成更复杂的程序化生成技术, 如基于物理的水力、风力侵蚀模拟, 以增强地形细节的真实

感。其次,引入动态纹理库和生态元素建模,提升场景的视觉丰富度。此外,将持续跟踪LLM技术的发展,探索其在理解更复杂、更具创造性地形描述方面的潜力,并进一步优化系统的实时性能,以拓展其在交互式应用中的实用性。

参考文献:

[1] YIN Z, WANG Y, PAPATHEODOROU T, *et al*. Text2VRScene: Exploring the framework of automated text-driven generation system for VR experience[C]//2024 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). Osaka, Japan: IEEE, 2024: 701-711.

[2] VALENCIA-ROSADO L O, STAROSTENKO O. Methods for procedural terrain generation: A review[C]//Mexican Conference on Pattern Recognition. Cham, Switzerland: Springer International Publishing, 2019: 58-67.

[3] FOURNIER A, FUSSELL D, CARPENTER L. Computer rendering of stochastic models[J]. Communications of the ACM, 1982, 25(6): 371-384.

[4] MUSGRAVE F K, KOLB C E, MACE R S. The synthesis and rendering of eroded fractal terrains[J]. ACM Siggraph Computer Graphics, 1989, 23(3): 41-50.

[5] CORDONNIER G, BRAUN J, CANI M P, *et al*. Large scale terrain generation from tectonic uplift and fluvial erosion[J]. Computer Graphics Forum, 2016, 35(2): 165-175.

[6] GAIN J E, MARAIS P C, STRABER W. Terrain sketching[C]//Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games (I3D '09). Boston, Massachusetts: ACM, 2009: 31-38.

[7] SPICK R, WALKER J A. Realistic and textured terrain generation using GANs[C]//Proceedings of the 16th ACM SIGGRAPH European Conference on Visual Media Production. London, United Kingdom: ACM, 2019: 1-10.

[8] 江西林, 王少荣, 李文玉, 等. 以特征线和高程范围为约束的DEM-cGAN框架[J]. 计算机辅助设计与图形学学报, 2021, 33(8): 1191-1201.

[9] BORNE P, CZERKAWSKI M, MARTIN R, *et al*. MESA: Text-driven terrain generation using latent diffusion and global Copernicus data[C]//2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Seattle, WA, USA: IEEE, 2025: 3058-3066.

[10] ZHANG G, CHEN L, ZHANG Y, *et al*. Translating words to worlds: Zero-shot synthesis of 3D terrain from textual descriptions using large language models[J]. Applied Sciences, 2024, 14(8): 3257.

[11] LIU J H, ZHANG S K, ZHANG C, *et al*. Controllable procedural generation of landscapes[C]//Proceedings of the 32nd ACM International Conference on Multimedia. Melbourne, Australia: ACM, 2024: 6394-6403.

[12] TIAN H. Is ChatGPT the ultimate programming assistant—How far is it[EB/OL]. (2023-04-24). https://arXiv preprint arXiv: 2304.11938.

[13] TAMKIN A, BRUNDAGE M, CLARK J, *et al*. Understanding the capabilities, limitations, and societal impact of large language models[EB/OL]. (2021-02-04). https://arXiv preprint arXiv: 2102.02503.

[14] BANG Y, CAHYAWIJAYA S, LEE N, *et al*. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity[EB/OL]. (2023-02-08). https://arXiv preprint arXiv: 2302.04023.

[15] BORJI A. A categorical archive of ChatGPT failures[EB/OL]. (2023-02-06). https://arXiv preprint arXiv: 2302.03494.

[16] LIU H, NING R, TENG Z, *et al*. Evaluating the logical reasoning ability of ChatGPT and GPT-4[EB/OL]. (2023-04-07). https://arXiv preprint arXiv: 2304.03439.

[17] ZALA A, CHO J, LIN H, *et al*. Envgen: Generating and adapting environments via LLMs for training embodied agents[EB/OL]. (2024-03-18). https://arXiv preprint arXiv: 2403.12014.

A Text-Driven Continuous Terrain Generation Method for Virtual Scenes Based on Large Language Models

WANG Weijia[1], LU Yuanjun[2], JI Bin[1], LI Jianhua[1], GUO Weibin[1]

(1. School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China; 2. Infosys Technologies (China) Co. Ltd, Hangzhou Branch, Hangzhou 310056, China)

Abstract: With the rapid advancement of three-dimensional technologies including game development, Virtual Reality (VR) and digital twins, there is an increasingly urgent demand for the automatic generation of high-quality, large-scale continuous virtual terrain scenes. Conventional manual interactive workflows involve cumbersome operations and high professional thresholds. Although procedural generation methods can improve efficiency, they still rely on graphical interfaces and the tuning of specialized parameters, making it difficult to flexibly capture users' semantic intentions. This paper proposes an automatic text-driven generation method for continuous virtual terrain based on the Large Language Model (LLM), which directly drives the generation pipeline through natural language to realize efficient construction of continuous virtual terrain scenes. A multi-stage task decomposition strategy is adopted in the proposed method to split the terrain generation pipeline into seven steps. By integrating prompt engineering and procedural generation techniques, an end-to-end mapping from text to large-scale continuous 3D terrain is established. To address the limitations of LLMs in complex tasks such as defects in multi-hop reasoning, output randomness and memory loss, mechanisms including cross-stage key information transmission, error retry and historical log learning are designed to improve the reliability and consistency of the system. A prototype system developed on the unity engine verifies the effectiveness of the proposed method. Test results on four typical terrain themes demonstrate that the Full Step System (FSS) achieves an average error rate of only 12.0%, which is significantly reduced by approximately 19–47 percentage points compared with the comparative systems without core mechanisms. In terms of efficiency, the average time consumption for generating a single preview terrain tile (1 024 × 1 024 vertex mesh) is about 42.14 s with controllable resource consumption. Further comparative experiments prove that the virtual terrain generated by the proposed method meets the expected requirements in generation efficiency, visual consistency and scalability. The proposed method remarkably improves generation efficiency and usability, offering a feasible solution for non-professional users to rapidly construct high-quality, large-scale 3D terrain via natural language.

Key words: human-computer interaction; text-driven; large language model; virtual terrain generation; procedural content generation

（责任编辑：张欣）