GenUSD: 3D Scene Generation Made Easy

Jiashu Xu
Yunhao Ge
Yifan Ding
Yin Cui
jiashux@nvidia.com
yunhaog@nvidia.com
yifand@nvidia.com
yinc@nvidia.com
NVIDIA
USA

Chen-Hsuan Lin
Xiaohui Zeng
Zekun Hao
Zhaoshuo Li
chenhsuanl@nvidia.com
xzeng@nvidia.com
zhao@nvidia.com
maxzhaoshuol@nvidia.com
NVIDIA
USA

Donglai Xiang
Qianli Ma
Fangyin Wei
J.P. Lewis
donglaix@nvidia.com
qianlim@nvidia.com
fangyinw@nvidia.com
jpl@nvidia.com
NVIDIA
USA

Qinsheng Zhang
Seungjun Nah
Arun Mallya
qinshengz@nvidia.com
snah@nvidia.com
amallya@nvidia.com
NVIDIA
USA

Jingyi Jin
Hanzi Mao
Yen-Chen Lin
jingyij@nvidia.com
hanzim@nvidia.com
yenchenl@nvidia.com
NVIDIA
USA

Pooya Jannaty
Tsung-Yi Lin
Ming-Yu Liu
pjannaty@nvidia.com
tsungyil@nvidia.com
mingyul@nvidia.com
NVIDIA
USA

ABSTRACT

We introduce GenUSD, an end-to-end text-to-scene generation framework that transforms natural language queries into realistic 3D scenes, including 3D objects and layouts. The process involves two main steps: 1) A Large Language Model (LLM) generates a scene layout hierarchically. It first proposes a high-level plan to decompose the scene into multiple functionally and spatially distinct subscenes. Then, for each subscene, the LLM proposes objects with detailed positions, poses, sizes, and descriptions. To manage complex object relationships and intricate scenes, we introduce object layout design meta functions as tools for the LLM. 2) A novel text-to-3D model generates each 3D object with surface meshes and high-resolution texture maps based on the LLM's descriptions. The assembled 3D assets form the final 3D scene, represented as a Universal Scene Description (USD) format. GenUSD ensures physical plausibility by incorporating functions to prevent collisions.

ACM Reference Format:
Jiashu Xu, Yunhao Ge, Yifan Ding, Yin Cui, Chen-Hsuan Lin, Xiaohui Zeng, Zekun Hao, Zhaoshuo Li, Donglai Xiang, Qianli Ma, Fangyin Wei, J.P. Lewis, Qinsheng Zhang, Seungjun Nah, Arun Mallya, Jingyi Jin, Hanzi Mao, Yen-Chen Lin, Pooya Jannaty, Tsung-Yi Lin, and Ming-Yu Liu. 2024. GenUSD: 3D Scene Generation Made Easy. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Real-Time Live! (SIGGRAPH Real-Time Live! '24), July 27 - August 01, 2024.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3641520.3665306

SIGGRAPH Real-Time Live! '24, July 27 - August 01, 2024, Denver, CO, USA

ACM ISBN 979-8-4007-0526-7/24/07
https://doi.org/10.1145/3641520.3665306

Figure 1: End to End text-to-scene generation results with prompt "A Japanese Garden."

1 INTRODUCTION

High-quality, large-scale 3D scenes are pivotal for content creation and training robust embodied agents. However, existing scene creation mostly depends on costly human annotations or 3D scans, which limit the scalability. Recent Large Language Models (LLM) assisted scene generation uses LLM planning to propose relevant objects and determine their locations via spatial reasoning. The layout is integrated with external assets by semantic matching and rendered in a physical simulator, e.g., Nvidia Omniverse for physical plausibility. However, this approach faces two major challenges: First, relying on LLMs for precise object placement is problematic, particularly in complex scenes involving numerous objects and intricate relationships. Furthermore, dependencies on external assets constrain the generalization as the rendered scenes are contingent upon the availability and compatibility of assets.

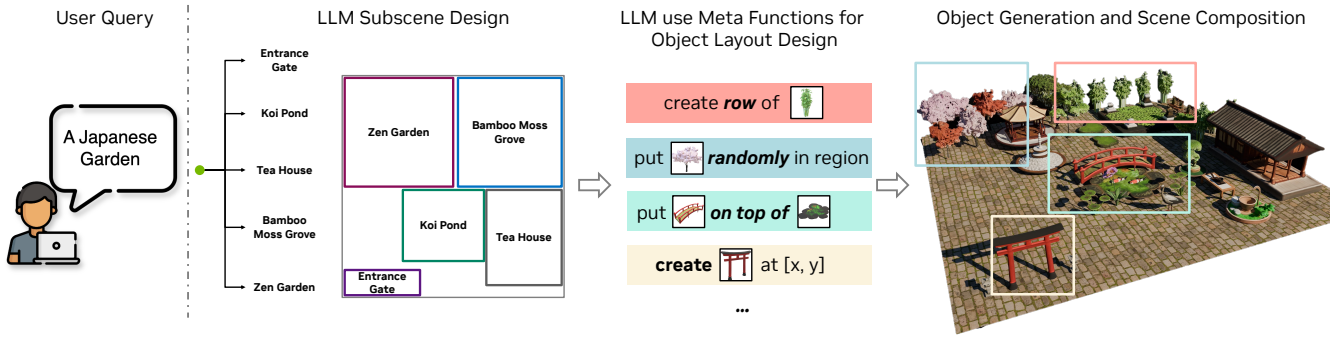

Figure 2: GenUSD can create complex, realistic scenes *without pre-defined assets*. Given a user query, an LLM decomposes the scene description into functionally and spatially distinct subscenes (Section 2.1.1), proposes relevant objects for each subscene (Section 2.1.2), and then generates size, position, and pose for each object via meta functions (Section 2.1.3). We use a novel text-to-3D model to generate objects on demand and use Omniverse for constructing physically plausible scenes (Section 2.2).

In this work, we introduce GenUSD, an end-to-end text-to-scene generation framework that uses the proposed object layout design meta functions for complex and physically plausible 3D scene generation, reducing reliance on LLM-generated precise locations. GenUSD employs a novel text-to-3D generation model to synthesize 3D assets with surface meshes and high-resolution texture maps, eliminating the need for external 3D assets. To the best of our knowledge, GenUSD is the first end-to-end text-to-scene generation method that uses only a simple text prompt as input to generate diverse high-quality 3D scenes with disentangled 3D objects.

2 END-TO-END TEXT TO SCENE GENERATION

Given a text query q describing the desired scene, GenUSD employs an LLM to generate a realistic and complex 3D scene that coherently aligns with the user query through a hierarchical approach (Figure 2). This process is fully end-to-end and independent of any external 3D assets, as each asset is created dynamically on demand. Examples of generated scenes are illustrated in Figure 1.

2.1 Hierarchically Scene Layout Generation

As shown in Figure 2, GenUSD hierarchically generates the scene layout, starting with a high-level decomposition into distinct subscenes (Section 2.1.1). It then generates detailed object proposals for each sub-scene (**??**), specifying pose, size, and position via LLM utilizing meta functions as tools Section 2.1.3).

2.1.1 **High-level Subscene Planning**. Even a simple query like "an industrial warehouse" can imply complex, unstated requirements. Real-world scenes may include hundreds of objects, from large items like carts and conveyor belts to small items like kits and boxes. To alleviate the planning burden on language models in generating all objects at once, GenUSD adopts a divide-and-conquer strategy decomposing the entire scene q into m functionally and spatially distinct subscenes $s_1, \ldots, s_m$. For instance, a query like "Cozy Bookstore Café" can be segmented into an entrance, main seating area, counter, and restrooms. This task decomposition significantly enhances the overall scene complexity and layout quality.

2.1.2 **Main Objects Proposal**. For each generated subscene s_i, GenUSD proposes relevant objects likely to be found in that subscene. For each subscene s_i, at least n relevant objects are generated, each with a detailed description including shape, color, and material information for subsequent 3D object generation (Section 2.2).

Preliminary experiments indicate challenges for LLMs in consistently generating objects of all sizes. Thus, GenUSD primarily generates larger objects while annotating smaller objects (with names only) on those larger objects if they have a supportive plane, e.g., tagging "pan", "cup", and "rag" on a "kitchen counter".

2.1.3 **Subscene Layout Generation via Meta function**. Following the object proposals, constructing a feasible object layout for each subscene remains challenging. Prior works can not handle complex scene layout generation with hundreds of objects. Inspired by the LLMs' excellent code-solving capacity, we propose prompting LLMs to utilize object layout meta functions. These meta functions encode the relational scene graph to design the layout, and their execution provides detailed pose and position for each object. Specifically, we develop a Python interface with custom classes and meta functions to determine the object location and pose (Figure 2). These meta functions address common requirements in object layout design, including: 1) object relationship constraints such as `next-to`, `face-to` and `on-top`. 2) geometric object layout patterns such as `row`, `grid`, and `circle`. Python's control loop provides the flexibility to design general meta functions for layout design. In summary, the LLM first uses the meta functions to create a high-level object layout design for the subscene. It then executes these meta functions to generate detailed object poses and positions.

2.2 Object Generation with Text-to-3D model

A downside in previous studies is their reliance on an external 3D asset pool (e.g., Objaverse) to populate the generated scene plan. The diversity of these external assets inherently constrains scene complexity. GenUSD addresses this shortfall by integrating a new text-to-3D generation model[1], which enable the synthesis of *any* novel objects by simply using a description of the object along with material specifications generated in Section 2.1.2. The resulting 3D assets have high-quality surface geometry and texture maps, and they are also highly faithful to the descriptions provided, obviating the need for any external asset inputs.

The generated assets, along with the optimized layout (Section 2.1.3), serve as the input of Omniverse, yielding a physically plausible scene configuration. Omniverse generates output in USD format, which is universally executable across multiple platforms.

[1]https://build.nvidia.com/shutterstock/edify-shutterstock-3d-txt23d-2pt7b