applied sciences

MDPI

Article

Translating Words to Worlds: Zero-Shot Synthesis of 3D Terrain from Textual Descriptions Using Large Language Models

Guangzi Zhang [1,*,†], Lizhe Chen [1,*,†], Yu Zhang [1], Yan Liu [1], Yuyao Ge [1,2,3] and Xingquan Cai [1]

[1] College of Information, North China University of Technology, Beijing 100144, China; zhangyu@mail.ncut.edu.cn (Y.Z.); liuyan@mail.ncut.edu.cn (Y.L.); yuyao.ge.work@gmail.com (Y.G.); caixingquan@ncut.edu.cn (X.C.)

[2] CAS Key Laboratory of AI Security, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

[3] University of Chinese Academy of Sciences, Beijing 101408, China

[*] Correspondence: guangzi@ncut.edu.cn (G.Z.); chenlizhe@mail.ncut.edu.cn (L.C.)

[†] These authors contributed equally to this work.

Abstract: The current research on text-guided 3D synthesis predominantly utilizes complex diffusion models, posing significant challenges in tasks like terrain generation. This study ventures into the direct synthesis of text-to-3D terrain in a zero-shot fashion, circumventing the need for diffusion models. By exploiting the large language model's inherent spatial awareness, we innovatively formulate a method to update existing 3D models through text, thereby enhancing their accuracy. Specifically, we introduce a Gaussian–Voronoi map data structure that converts simplistic map summaries into detailed terrain heightmaps. Employing a chain-of-thought behavior tree approach, which combines action chains and thought trees, the model is guided to analyze a variety of textual inputs and extract relevant terrain data, effectively bridging the gap between textual descriptions and 3D models. Furthermore, we develop a text–terrain re-editing technique utilizing multiagent reasoning, allowing for the dynamic update of the terrain's representational structure. Our experimental results indicate that this method proficiently interprets the spatial information embedded in the text and generates controllable 3D terrains with superior visual quality.

Keywords: text-to-3D; multiagent; chain-of-thought; large language model; LLM's spatial awareness; terrain synthesis

check for updates

Citation: Zhang, G.; Chen, L.; Zhang, Y.; Liu, Y.; Ge, Y.; Cai, X. Translating Words to Worlds: Zero-Shot Synthesis of 3D Terrain from Textual Descriptions Using Large Language Models. *Appl. Sci.* **2024**, *14*, 3257. https://doi.org/10.3390/app14083257

Academic Editor: Andrea Prati

Received: 17 March 2024
Revised: 10 April 2024
Accepted: 10 April 2024
Published: 12 April 2024

1. Introduction

Recent initiatives like Dream3D [1] have marked considerable progress in translating text into 3D models, managing to create complex and stylistically varied 3D entities under zero-shot conditions. However, these advancements typically depend on 2D images generated by diffusion models [2], leading to outputs with limited precision control. This limitation becomes particularly pronounced in tasks that demand extensive scope and detail, such as 3D terrain generation. In these scenarios, the existing methods often falter in accurately interpreting spatial information, including direction and location, from text inputs. Furthermore, the task of making precise modifications to pre-existing 3D models through text commands remains a formidable challenge in the realm of diffusion-model-based techniques, as these methods generally lack the finesse to enact localized alterations on 3D structures based purely on textual directives.

Diverging from the standard text-to-3D paradigms, our approach eschews the use of diffusion models and multiview synthesis, the typical intermediaries. Instead, it directly leverages large language models. Although these models are adept at parsing textual data, their proficiency in decoding spatial information—such as instructions to "generate a bouquet of flowers in the top-left corner of the image" or "draw a sea adjacent to this volcano"—is notably limited. This shortfall often results in inaccuracies when they are directly employed for crafting complex and expansive terrain datasets, occasionally yielding outputs that starkly deviate from the intended text descriptions. The architecture of our method is delineated in Figure 1.

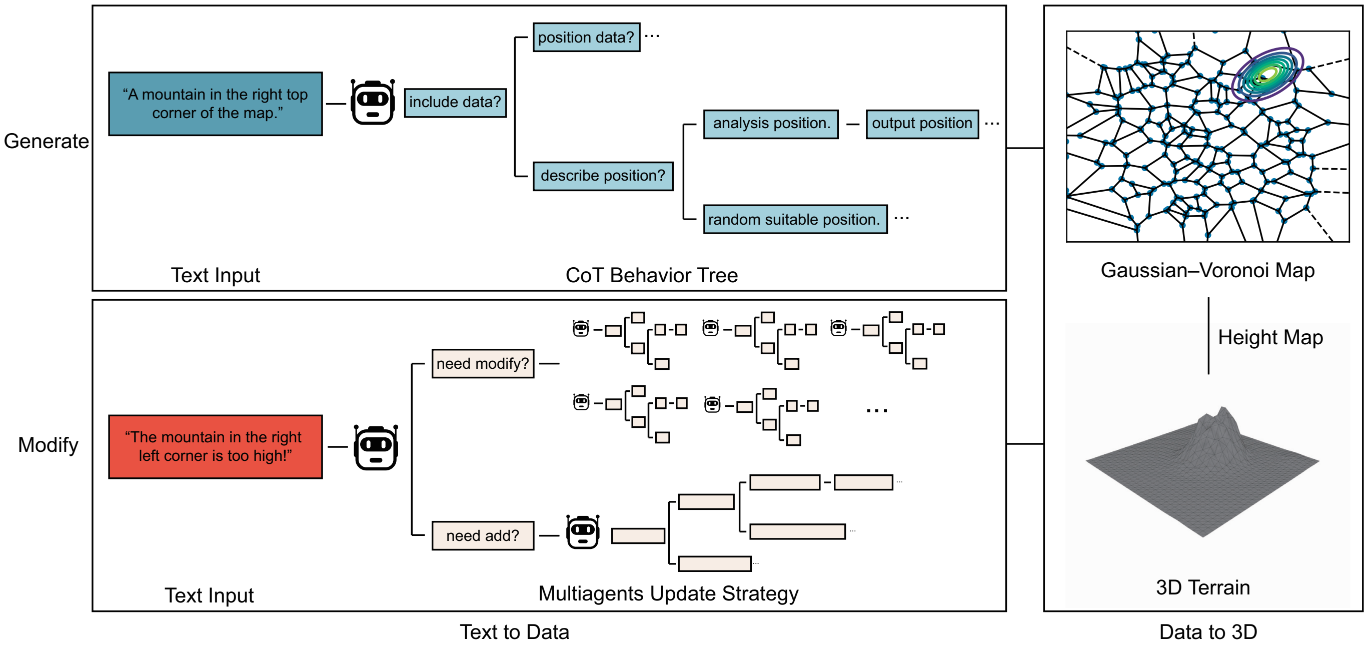


Figure 1. In summary, our method mainly consists of two aspects of design. The first aspect is the text-to-data process for capturing data from text, which is primarily achieved by the chain-of-thought behavior tree and Multiagents update strategy. On the other hand, the data-to-terrain process is implemented mainly by the Gaussian–Voronoi map. This process completely abandons the diffusion model, allowing us to generate and edit our 3D models more accurately.

To surmount this challenge, we employ "heightmaps" as a foundational tool for terrain depiction [3] and introduce a novel terrain description data structure: the Gaussian–Voronoi map. This structure is designed to generate complex and varied 3D terrains from minimalistic input data. The Gaussian–Voronoi map processes simple terrain descriptions using Gaussian functions, efficiently morphing them into intricate terrains. Subsequently, we developed a chain-of-thought behavior tree method. This novel approach, anchored in thought chains and behavior trees, steers the large language model in distilling terrain features from textual inputs and translating these features into standardized terrain information data. This information then informs the Gaussian–Voronoi map to produce preliminary 3D terrains. During the terrain detail adjustment phase, we deployed a multiagent system, with independent agents assigned to specific terrain features on the Gaussian–Voronoi map. These agents dynamically refine the 3D terrain based on textual instructions, thereby enhancing the method's adaptability and utility.

Our experimental findings affirm that our approach not only grasps the spatial information contained in the input texts but also generates high-quality 3D terrains that fulfill diverse editing requirements and surpass the performance of conventional text-to-3D generation methods.

In summary, our contributions are as follows:

1. Introduction of the Gaussian–Voronoi map, an innovative expression of 3D terrain, capable of receiving simple terrain descriptions in Gaussian function format and transforming them into detailed 3D terrains;
2. A chain-of-thought behavior tree method for large language models, enabling the systematic extraction and quantification of precise terrain features from varied textual inputs into data with a standardized scale;

3. A multiagent-based text-to-terrain modification strategy that effectively fulfills terrain modification needs from textual inputs, ensuring coherent terrain evolution through the deployment of multiple agents.

2. Related Works

2.1. Text-to-3D Generation

The development [4–6] of diffusion models and high-quality novel view synthesis technologies has recently made generating 3D models a hot topic. The usual approach involves using diffusion models to generate 2D images, building an intermediate bridge from text-to-3D models, and then using novel view synthesis methods to generate 3D models from images. Some studies [7–11] attempt to express 3D models in the form of implicit neural radiance fields, while other methods [12–21] focus on more traditional forms, such as voxels, point clouds, and meshes. One important thing is, benefiting from the excellent 3D scene synthesis quality provided by recent 3D Gaussian splatting, researchers have proposed methods [22–24] for generating 3D models with higher quality or performance. The biggest challenge faced by these methods is the instability of obtaining multiview image information from diffusion models. In response to this problem, diverse solutions have been proposed: some methods [25,26] guide the diffusion model to generate images with better consistency in the form of image priors; others have achieved zero-shot 3D synthesis by introducing CLIP [27,28] technology. Among them, Dream3D, by introducing a 3D shape prior mechanism, uses a prior model as an initializer for the neural radiance field and gradually optimizes it with a full prompt, achieving high-quality text-to-3D effects.

2.2. Large Language Models

The advent of large language models (LLMs), such as GPT-4, has introduced their reasoning capabilities into decision making across various fields. Scholars and experts have started to study the reasoning mechanisms and action methods in language models, such as React [29] and Self-Refine [30]. Among them, the chain-of-thought [31,32] technique has proven that, without any additional training, simply structuring the input text to the large language model can guide it to better understand the input text.

Beyond research related to prompt engineering, people have also explored the capabilities of large language models in addressing various problems. Some studies [33,34] have pointed out that the reasoning ability of large language models is significantly enhanced when abstract problems are referred to with real-world scenarios, while some other studies [35–37] also find that large language models still have potential superior reasoning capabilities compared with traditional language models even when facing abstract problems such as finding the shortest path.

Moreover, prior to this paper, it was generally believed that large language models have poor spatial awareness and struggle with spatial geometric problems. Our method explores the reasoning quality of LLMs in Euclidean space scenarios from another angle, powerfully demonstrating the immense potential of LLMs in spatial perception.

2.3. Agent Technologies Based on Large Language Models

Humans have always pursued artificial intelligence that is equivalent to or surpasses human intelligence, and agents are considered an effective means to achieve this pursuit. Recent works have started creating agents with advanced planning and judgment capabilities using large language models. Among them, some studies [38–41] focus on enhancing the problem-solving capabilities of multiagent systems, such as Stable-Alignment, which creates a guiding dataset through interactions between LLM agents, strengthening the perception and collaboration capabilities among multiple agents; the Dynamic LLM-Agent Network proposes a strategic team of agents method, DyLAN, allowing multiple agents to interact within a task-based dynamic framework. Another part of the research [42–44] focuses more on simulating social behaviors with multiple agents, such as generative agents, which created a virtual human community led by agents, successfully simulating some aspects of human society's operation; language agents with reinforcement learning use agents to simulate strategic role playing in the Werewolf game.

In our research, we equip agents with short-term memory and abstract tasks for the real-time secondary editing of terrain features, opening new avenues for agent applications.

3. Method

We introduce a novel method to generate complex 3D terrain models. This method overcomes the limitations of existing diffusion-model-based approaches, which struggle with large and complex terrains. This difficulty arises because diffusion models have limited spatial awareness, such as understanding directional information (north, south, east, west) within images, and cannot convert text describing terrain orientation into data. Additionally, this limitation makes it challenging to make localized changes to already-generated models using only text. Our method leverages the spatial awareness capabilities of large language models, bypassing diffusion models, and is divided into three steps: First, we design a Gaussian–Voronoi map data structure to generate complex terrain heightmaps from simple data inputs. Second, we construct a chain-of-thought behavior tree strategy to extract terrain data from text inputs. Lastly, we introduce a multiagent-based method for terrain feature adjustment to support detailed editing and updating of generated terrains through text.

3.1. Gaussian–Voronoi Map

The primary challenge in generating terrain data using large language models is the lack of stability when outputting large amounts of text, making it difficult to directly output complex terrain data. To tackle this challenge, we introduce the Gaussian–Voronoi map, an innovative data structure. This method combines the spatial partitioning capability of Voronoi diagrams with the smoothing characteristics of Gaussian functions to simulate diverse terrain features. Figure 2 explains the working principle of the Gaussian–Voronoi map.

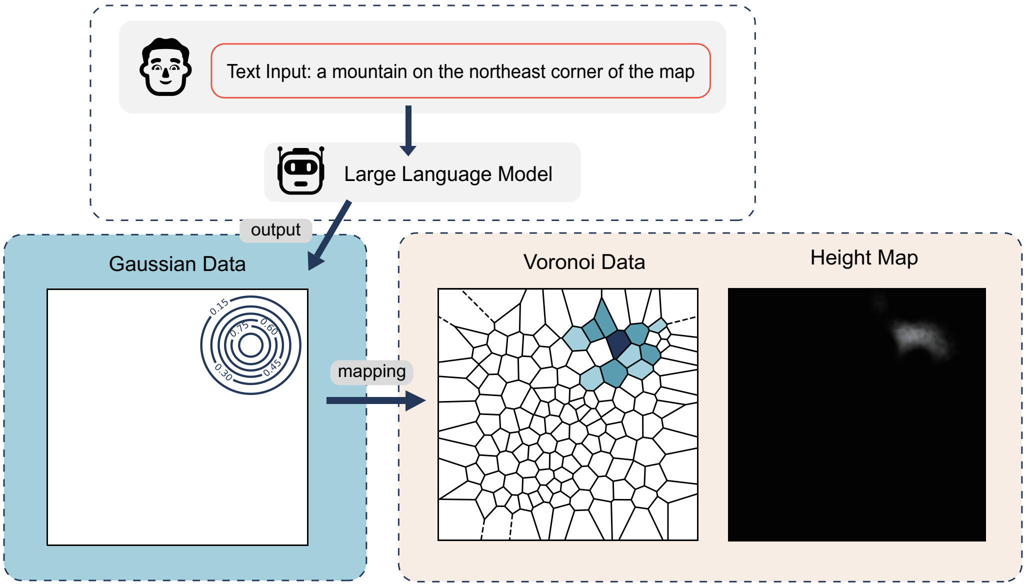


Figure 2. Significant differences exist between the cells of the original Voronoi diagram. To ensure that terrain generation is roughly equivalent across the map, we apply Lloyd's relaxation to the Voronoi diagram. This maintains a basic uniformity in cell sizes.

Specifically, we first use a Gaussian function to roughly express the height of a certain area:

$$G = h \cdot e^{-\frac{d^2}{w}} \tag{1}$$

where h represents the elevation height represented by the Gaussian function, w is the influence weight of the Gaussian function on the map, and d is the distance from any point on the map to the center of the Gaussian function. Diverging from traditional Gaussian applications, we avoid using a covariance matrix for scaling and rotation control. Instead, we add multiple Gaussian functions to simulate these effects. This simplification helps the large language model process terrain features more efficiently, reducing potential errors. Instead, we achieve a similar effect by adding more Gaussian functions on the map. The advantage of this method is that, since the parameters of the Gaussian functions are generated by a large language model, omitting control over rotation and scaling simplifies the dimensions that the model must handle when analyzing terrain features, thereby effectively reducing the potential error rate. This strategy aims to leverage the capabilities of large language models while alleviating their burden in handling complex terrain generation tasks.

After generating Gaussian data, we integrate them into a Voronoi diagram. This step divides the space into regions, each influenced by a Gaussian peak, creating a more varied terrain. We use Voronoi diagrams to subdivide the plane into several approximately equal-sized areas. For any pixel point on the plane, we find the nearest Voronoi point p_i, thus determining the Voronoi area to which the pixel point belongs; next, for each pixel point corresponding to p_i, we shape the terrain details by overlapping Gaussians. To capture the terrain's randomness and diversity, we go beyond letting each Gaussian impact the terrain uniformly. We employ a probability function, $c_{i,j}$, to decide if a Gaussian G_j affects a specific point, enhancing the model's realism and flexibility. The probability function is determined by two Gaussian distribution functions:

$$c_{i,j} = e^{-\frac{d_j^2}{n w_j}} \tag{2}$$

where d_j, w_j, and n have the same meanings as in the previous formula, with n being the L2 norm of the map size. For all G_j determined to be superimposed on p_i, we simply sum their values to obtain the value of that block:

$$H(p_i) = \sum_{j=1}^{k} G_j \tag{3}$$

Finally, rendering each block according to its data into a grayscale image yields the 3D terrain heightmap. By inputting data to the large language model in the form of Gaussians and decoding them through the Gaussian–Voronoi map structure, we can convert simple summaries of terrain features into complex terrains; by adjusting the number of Voronoi blocks, we can obtain 3D terrain heightmaps of specified size and complexity. We chose Gaussian as the container for simple terrain features mainly because Gaussian can summarize the general terrain height with just four parameters, as mentioned above: the two-dimensional coordinates, elevation height, and influence weight, allowing the large language model to describe and adjust 3D terrains with high accuracy. On the other hand, the Gaussian construction is simple, clear, and common, reducing the difficulty of describing the necessary data content and its significance to the large language model. The strong advantages brought by Gaussian enable us to achieve high-quality data output even with zero-shot, which will be detailed in the experimental section. The choice of Voronoi for converting simple data to complex data is due to its ability to divide the entire image with almost equal weight and its rich randomness, which can better simulate 3D terrain scenes, as has been fully verified in some game-making methods before.

However, directly using the Gaussian–Voronoi map to generate heightmaps causes some problems: in many situations, this generation method may lead to overly abrupt height changes. The root cause of this problem is the superposition of multiple Gaussians, which may create significant height differences in specific areas compared with surrounding areas.

To solve this problem, we adopted a simple and effective method, namely, directly applying Gaussian blur on the Gaussian–Voronoi map. Although there was initial concern that this might cause terrain features with large height differences, such as cliffs, to appear smooth, in practice, we found that the height differences shaped by these terrain features themselves were sufficient to offset the smoothing effect of the blur. At the same time, the height differences in exceptional areas are not easily affected by the blur, so this method has shown considerable robustness in practice. Figure 3 shows the effect of using Gaussian blur.

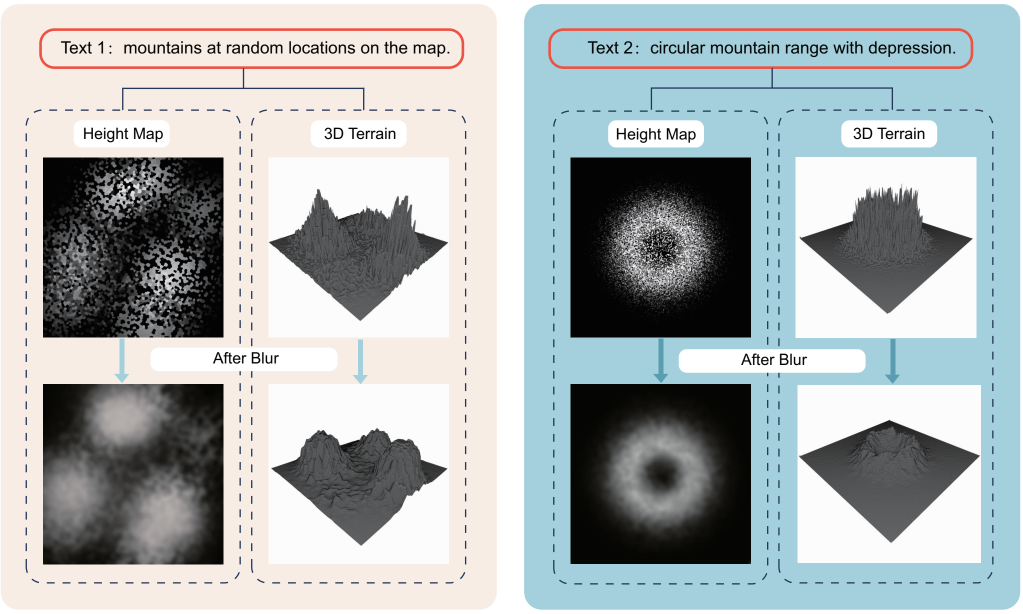

Text 1 : mountains at random locations on the map.
Height Map
3D Terrain
After Blur
Text 2 : circular mountain range with depression.
Height Map
3D Terrain
After Blur

Figure 3. The essence of Gaussian blur is the elimination of excessive high-frequency information contained in our heightmap. This occurs during the mapping from Gaussian to Voronoi, and is difficult to avoid at this stage, as our fundamental purpose is to extract as much high-frequency information as possible from the ultra-low-frequency information in Gaussian, thereby enhancing the richness and complexity of the random terrain.

Through this simple and ingenious processing method, we successfully mitigated the issue of overly abrupt height changes caused by the superposition of multiple Gaussians, improving the smoothness of the generated heightmaps while maintaining a reasonable representation of terrain features. This adjustment not only effectively enhanced the quality of the generated results but also ensured that important details of the original terrain were not lost in the blurring process.

3.2. Chain-of-Thought Behavior Tree

Our goal is for large language models to process various types of text inputs and extract sufficient spatial information. This includes directly requesting the generation of specific geographical structures at certain coordinates or indirectly describing the orientation and partial features of terrains. To achieve this, the model must perform a degree of reasoning on the text inputs. This requires the large language model to perform a certain degree of reasoning on the text inputs. To accurately interpret the terrain and spatial information contained within the input texts, a common method involves guiding the large language model through a thought chain to think in steps according to a set logic, and extensive research has demonstrated the effectiveness of this approach. However, considering the diversity of text inputs, guiding the large language model to recognize text content in a fixed pattern still cannot accurately and effectively process various inputs.

We developed a zero-shot guidance method, named the chain-of-thought behavior tree. This method combines thought chains and behavior trees to guide the large language model. It processes data using different reasoning logics for analyzing terrain feature information. For each input text, we engage the large language model in multiple rounds of interaction. In each round, we present guiding text in a fill-in-the-blank format. The model is tasked with completing these blanks based on the input text. The answers to these fill-in-the-blanks are then used as criteria in the behavior tree to determine which logical path the large language model should follow in subsequent guidance. To maintain continuity and consistency of context, in subsequent guiding inputs, we not only provide new guiding texts but also include the answers formed by the large language model in the previous judgment process, thus achieving a step-by-step reasoning process.

Figure 4 illustrates a simple model of the chain-of-thought behavior tree. Here, the large language model first checks if the input text includes data for constructing Gaussians. If so, the data are directly cached; otherwise, it begins to build data step-by-step. In the data construction process, to ensure the consistency of data measurement scales, we have the large language model first convert text information from diverse natural languages into specific geographical concepts, and then generate data from these concepts according to the given scale standards.

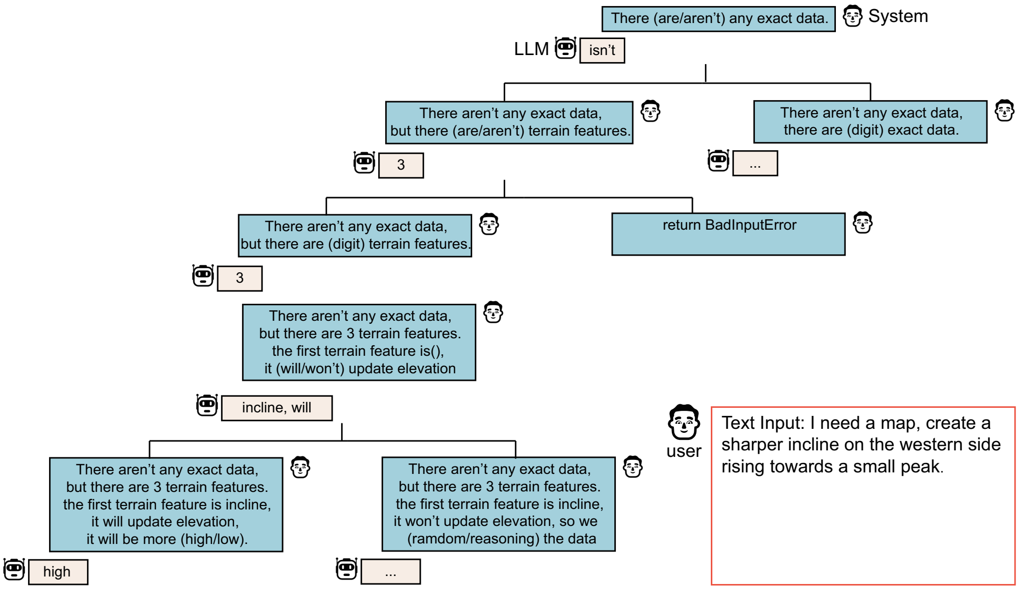

There (are/aren't) any exact data.
System
LLM
isn't
There aren't any exact data, but there (are/aren't) terrain features.
There aren't any exact data, there are (digit) exact data.
3
...
There aren't any exact data, but there are (digit) terrain features.
return BadInputError
3
There aren't any exact data, but there are 3 terrain features. the first terrain feature is(), it (will/won't) update elevation
incline, will
user
Text Input: I need a map, create a sharper incline on the western side rising towards a small peak.
There aren't any exact data, but there are 3 terrain features. the first terrain feature is incline, it will update elevation, it will be more (high/low).
There aren't any exact data, but there are 3 terrain features. the first terrain feature is incline, it won't update elevation, so we (ramdom/reasoning) the data
high
...

Figure 4. Similar research exists in related fields, where our method shares certain similarities with the thought-of-tree strategy. Unlike ToT and other LLM tree-reasoning methods that require LLMs to evaluate the current state on their own and use thought generators to automatically generate solutions, which allows the large language model to offer more ideas for unknown problems, suitable for complex situations with uncertain text domain problems, our method directly fixes the LLM's thought path without the need for additional solutions. This is because, in our scenario, the final problem to be solved has been fixed as "extracting terrain feature data from input text information", so we do not need LLMs to diverge too much but rather to gradually deduce and refine data based on the given solutions. Even in the final step of data generation, since our method can continue to edit the 3D terrain subsequently, there aren't any exact data required for LLMs to provide the best solution in one step. Compared with ToT, our method offers better controllability and lower performance overhead, as the number of times text information is input to LLMs is significantly reduced.

The chain-of-thought behavior tree method significantly enhances the large language model's ability to refine data from diverse input data when solving a single problem. Compared with a single thought chain, the behavior tree can dynamically adjust the reasoning path based on different features of the input, allowing the model to better adapt to different problems and scenarios, which is very important in our specified zero-shot scenario because it cannot be assumed for all types of problems. At the same time, by providing a clear structure of behavior trees and execution of thought chains, we offer clear decision-making and reasoning paths, making the model's decision process more transparent and traceable, thereby enhancing the interpretability of the model's output. By combining the Gaussian–Voronoi map and chain-of-thought behavior tree, the model is endowed with sufficient spatial awareness to generate diverse high-quality initial terrains. In the next section, we will discuss how to use text to make secondary modifications to the initial terrain.

3.3. Multiagent Update Strategy

In our study, we introduce a refined methodology to enhance the 3D terrain generation process using a new multiagent strategy. This strategy stems from the recognition that a 3D map embodies an ensemble of distinct terrain features, which are depicted using Gaussian functions within our Gaussian–Voronoi map framework. Our approach aims to facilitate the meticulous manipulation of specific map regions through the calibration of existing Gaussian functions or the introduction of new ones.

Upon the initial rendering of the 3D landscape, we implement a hierarchical multiagent system to refine and perfect the terrain. The process commences with the submission of modification directives and Gaussian parameters to the overseeing agent of agents. This pivotal agent assesses the overall terrain configuration to decide whether to engage the add or modify strategy.

When the add strategy is activated, the agent of agents, utilizing a bifurcated chain-of-thought behavior tree, initiates the process by incorporating additional Gaussians into the extant map schema. On the other hand, the modify strategy initiates with each agent conducting a preliminary hypothetical modification of their assigned Gaussian, independent of other Gaussians. The initial phase, or the first branch of their dedicated chain-of-thought behavior tree, enables an isolated assessment of potential adjustments. It does so without the influence of neighboring terrain features. Following this provisional step, the agents embark on the second part of their evaluation, the second branch of the behavior tree. Here, they scrutinize the outcomes of the assumed modifications in the context of the collective Gaussian landscape. This comprehensive analysis enables agents to discern the interplay between their Gaussian and adjacent ones, leading to a more informed and precise calibration of the terrain. Such a sequential approach ensures that any Gaussian whose modified presence is virtually nonexistent—whose influence on the terrain's relief is minimal to none—is systematically removed. This two-part chain-of-thought process not only refines individual features but also harmonizes the collective adjustments, resulting in an intricately detailed and cohesive 3D terrain.

Both strategies are operable in tandem, with precedence given to the add strategy. This precedence ensures that modifications applied during the modify phase can refine the Gaussians introduced in the add phase, resulting in a more accurate and textually compliant terrain outcome. Figure 5 shows our agent workflow.

This solution effectively introduces a secondary editing mechanism into our method, providing the ability to update and optimize the generated results, which is crucial for meeting specific requirements and a detailed design. Additionally, since the size of Gaussians is controllable, this approach also offers the possibility of fine-grained control over the terrain, allowing us to generate 3D terrains with infinite precision. An intuitive flaw in this method is that the same terrain feature represented by multiple Gaussians might be modified by multiple agents simultaneously, potentially leading to consistency issues in data measurement scales. However, we have already ensured the consistency of data scale processing by the large language model with the chain-of-thought behavior tree, so this issue does not require special consideration here.

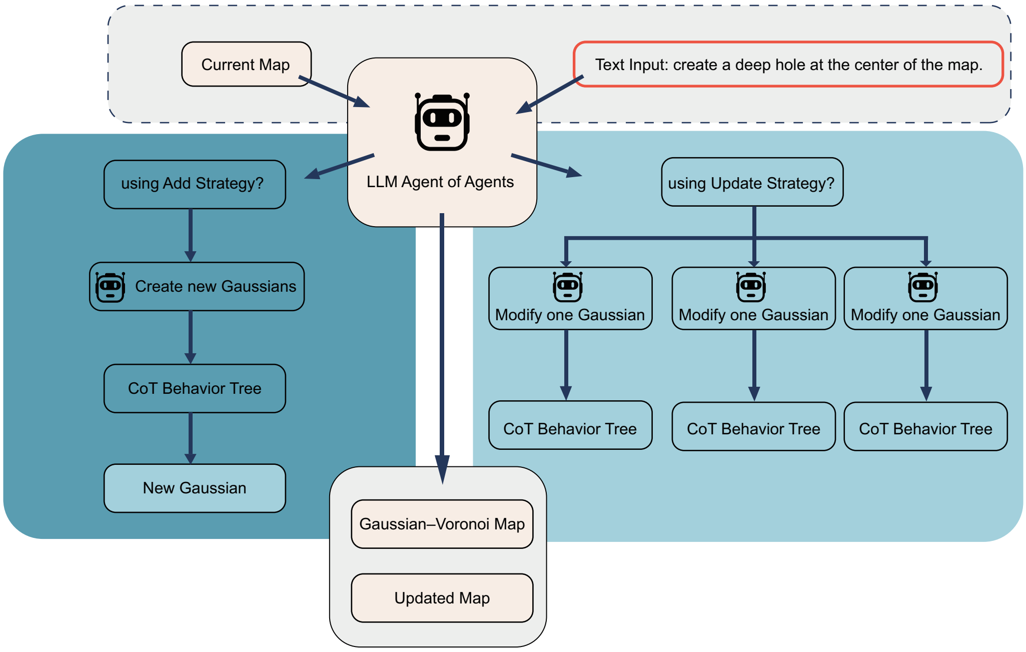


Figure 5. Unlike conventional multiagent systems, our agents do not have long-term memory. Instead, each time the terrain is modified, their memory only contains data from the last modification of the terrain feature, as longer-term memory is meaningless for the next modification the agent will make, and the only possible use of such memory, to undo the last change, can be completely bypassed without an LLM.

4. Results and Evaluation

In this chapter, we evaluate the effectiveness of our proposed text-to-terrain method. We first introduce the preparation before the experiments, including the hardware environment, evaluation metrics, and datasets. Then, we compare our method with the current state-of-the-art techniques, demonstrating the superior quality of our approach. Finally, we conduct ablation studies to prove the effectiveness of each step in our method.

4.1. Preparation

4.1.1. Hardware Configuration

All experimental results were produced on an Nvidia RTX 3090 Ti GPU. To avoid errors due to hardware differences, we did not directly use the data from the comparison methods' papers. Instead, we downloaded the open-source implementations of these methods and ran them in our local environment to obtain experimental data.

4.1.2. Evaluating Indicators

Given the unique nature of our generated models and the limitations of diffusion-model-based methods in spatial understanding, we found conventional metrics like R-precision or FID less applicable. This necessitated alternative evaluation approaches. This difficulty arose due to the lack of corresponding real-world models for our generated models and the limited spatial understanding of diffusion-model-based methods. To navigate this issue, we drew inspiration from subjective evaluation methods utilized in research [45] facing similar challenges. We devised a subjective assessment criterion tailored to our method. We engaged 30 participants to rate a range of 3D models on a 0 to 5 scale. To ensure scoring consistency, we provided detailed guidelines on how to match models with textual descriptions. Scores were allocated based on how well models matched textual descriptions: a score of 2 indicated that the model met some aspects of the text, 3 suggested that the model largely matched the text, and 4 implied that the model perfectly aligned with the text description.

4.1.3. Datasets

It is important to note that our dataset comprises solely textual input sequences; it does not include images for validation purposes. This is because our methodology does not necessitate any form of retraining of the model.

We directly used the large language model to generate a series of text data in various forms as the dataset, including the following:

- Simple sentence text descriptions (easy), e.g., "A plain with flat terrain", "A soft island".
- Text descriptions containing descriptions of the map and some geographical concepts (medium), e.g., "A mountain in the center of the map", "four wonderful mountains, on the four corners of the map".
- Text descriptions containing a large number of geographical concepts and descriptions (hard), e.g., "an island with a central plateau. Include a gentle slope on the eastern side leading down to a flat area, and a sharper incline on the western side rising towards a small peak".
- Text descriptions directly specifying geographical features in numerical form (digit).

The datasets mentioned above have been made public at our project homepage. Due to the limited space in this paper, we do not describe in detail the specific methods for generating them and their properties here. We have placed this content in the link mentioned above.

4.2. Quality Evaluation Experiments

Our evaluation experiments were designed to test the method's quality in three scenarios: without using an update strategy, with a single-step update, and with a five-step update process. Detailed criteria defined each update step's scope and impact on the model's performance. This approach was designed to validate the effectiveness of the multiagent update strategy. Through these tests, we compared our method against other leading-edge techniques, including DreamFusion, PureClipNeRF, and DreamGaussian, utilizing the CLIP models from OpenAI for our evaluations.

Figure 6 shows some of our experimental results on multiple datasets.

In the easy dataset, our no-update method outperformed PureClipNeRF and DreamGaussian by 0.2 and showed a notably larger advantage over DreamFusion by 0.6, as quantified by the adjusted scores in Table 1. This can be attributed to the smaller volume of input text and the relatively vague requirements for terrain generation in this dataset, where the unique spatial perception capabilities of our method were not fully leveraged. However, even with minimal textual input, our approach yielded higher-quality generations than those based on diffusion models.

Table 1. Quality assessment experiment results on the easy dataset.

Method	Regular Score	Adjusted Score [1]
DreamFusion	3.54	3.63
PureClipNeRF	3.84	3.82
DreamGaussian	4.01	4.03
Ours (without modify)	4.15	4.17
Ours (with 1 step modify)	4.23	4.26
Ours (with 5 steps modify)	4.52	4.69

[1] Adjusted score is calculated by removing the highest and lowest scores before averaging.

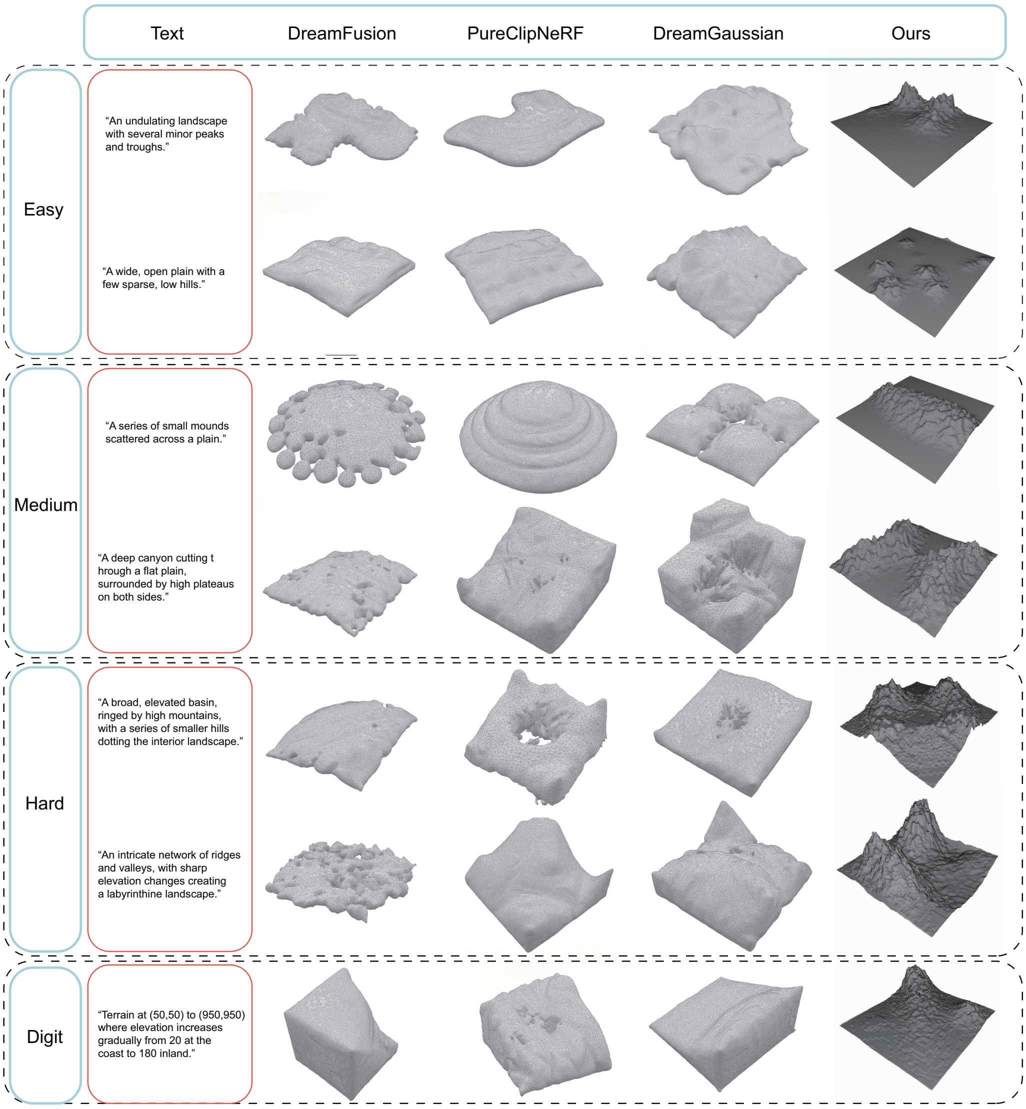

Text
DreamFusion
PureClipNeRF
DreamGaussian
Ours
Easy
"An undulating landscape with several minor peaks and troughs."
"A wide, open plain with a few sparse, low hills."
Medium
"A series of small mounds scattered across a plain."
"A deep canyon cutting t hrough a flat plain, surrounded by high plateaus on both sides."
Hard
"A broad, elevated basin, ringed by high mountains, with a series of smaller hills dotting the interior landscape."
"An intricate network of ridges and valleys, with sharp elevation changes creating a labyrinthine landscape."
Digit
"Terrain at (50,50) to (950,950) where elevation increases gradually from 20 at the coast to 180 inland."

Figure 6. We converted NERF and Gaussian results into mesh objects using methods from existing research [46,47]. This conversion facilitates a more effective comparison with our method in practical scenarios. It is observable that, compared with other methods based on diffusion models, our approach can leverage the language recognition capabilities of large language models to make accurate judgments about spatial information and generate precise three-dimensional terrains that align with the descriptions of location information. Additionally, our method employs heightmaps for construction, which significantly reduces the number of tiles required for the three-dimensional terrain, further enhancing the practical value of our approach.

We believe that short texts do not provide sufficient spatial information. Large language models can establish a mapping from short to extended texts, thus enriching the text with adequate spatial information. In contrast, diffusion models, which learn from two-dimensional images, struggle to grasp the association between spatial and textual information and cannot directly supplement more information in the form of extended text.

Our method's superior performance on the medium and hard datasets, with an improvement of 0.99 over the closest competitor, can be attributed to these datasets' complexity, as shown in Tables 2 and 3:

Table 2. Quality assessment experiment results on the medium dataset.

Method	Regular Score	Adjusted Score
DreamFusion	2.89	2.91
PureClipNeRF	2.96	2.93
DreamGaussian	2.84	2.87
Ours (without modify)	3.87	3.89
Ours (with 1 step modify)	3.98	4.01
Ours (with 5 steps modify)	4.32	4.26

Table 3. Quality assessment experiment results on the hard dataset.

Method	Regular Score	Adjusted Score
DreamFusion	2.03	2.12
PureClipNeRF	2.59	2.43
DreamGaussian	2.54	2.61
Ours (without modify)	3.53	3.57
Ours (with 1 step modify)	3.75	3.82
Ours (with 5 steps modify)	4.05	3.97

This complexity better showcases our approach's spatial perception capabilities, particularly in basin scenarios in Figure 1. The medium dataset introduced descriptions of spatial and orientation information, challenging for diffusion models due to their difficulty in maintaining accurate directionality across multiple input images. Consequently, diffusion models struggled with tasks requiring precise control over geometric space as specified in the text inputs.

Experimental results suggest that even when the text provides sufficient spatial information, diffusion models still struggle to map it onto image space. We believe this is because diffusion models tend to extract object information from the text and often overlook attributes like elevation. A potential improvement for diffusion models could involve training them on a large corpus of text-elevation map data for terrain synthesis. However, given the inherent complexity and variability of terrain, developing a specialized diffusion model that performs well might be challenging. This further highlights the limitations of using diffusion models alone for three-dimensional spatial perception and generation, and the superiority of our method.

As shown in Table 4, in the digit dataset, our method's ability to accurately process numerical descriptions of spatial features led to its overwhelming superiority, highlighting the effectiveness of the chain-of-thought behavior tree in handling direct numerical inputs for precise 3D terrain generation. Diffusion models typically fail to process features described numerically, especially when numbers are associated with spatial concepts. In contrast, our method benefits from the complexity of the chain-of-thought behavior tree and the limitations of large language models in processing thought. Direct numerical inputs provided a more reliable standard for the large language models, enabling the generation of 3D terrain models with higher accuracy and fidelity to the textual descriptions.

Table 4. Quality assessment experiment results on the digit dataset.

Method	Regular Score	Adjusted Score
DreamFusion	1.67	1.72
PureClipNeRF	2.01	1.95
DreamGaussian	1.68	1.74
Ours (without modify)	4.74	4.79
Ours (with 1 step modify)	4.83	4.77
Ours (with 5 steps modify)	4.81	4.86

The data presented above further demonstrate the efficacy and utility of our terrain editing approach based on the multiagent update strategy. This method allows for extensive updating of the terrain even when the initial generation results are not optimal, leading to significantly improved experimental outcomes. The adaptability of our approach is such that, without imposing limits on the number of adjustments, our method can generate infinitely refined 3D terrain content under any circumstances.

4.3. Ablation Experiments

Our ablation study aimed to delineate the specific impact of the Gaussian–Voronoi map and the chain-of-thought behavior tree on the terrain generation quality. We sought to understand how each component contributes to the realism and complexity of the generated terrains. This investigation aimed to evaluate the influence of the Gaussian–Voronoi map, the chain-of-thought behavior tree, and the multiagent update strategy on the proficiency of generating terrain from textual descriptions. For the purposes of this study, we chose to exclude the multiagent update strategy. This decision was informed by insights gained from previous quality assessment experiments, which convincingly demonstrated the benefits of incorporating this strategy. As a result, by omitting the multiagent update strategy from our ablation study, our goal was to more precisely discern the effects of the Gaussian–Voronoi map and the chain-of-thought behavior tree on enhancing the quality, authenticity, and intricacy of the generated terrain models. This refined focus enabled us to draw more direct correlations between these components and the overall success of our terrain generation method.

Figure 7 visually demonstrates the multiagent update strategy's role, highlighting its collaboration with other components to enhance terrain generation's efficiency and accuracy. This strategy's integration is pivotal for refining terrain details and aligning them closely with textual descriptions. This inclusion ensures that readers can fully appreciate how the multiagent update strategy collaborates with other key components to facilitate an efficient and accurate process of generating terrain from textual descriptions. By doing so, we underscore that although the multiagent update strategy was not considered in this specific analysis, it plays a crucial role within the overall framework, being instrumental in achieving the ultimate objectives of our method.

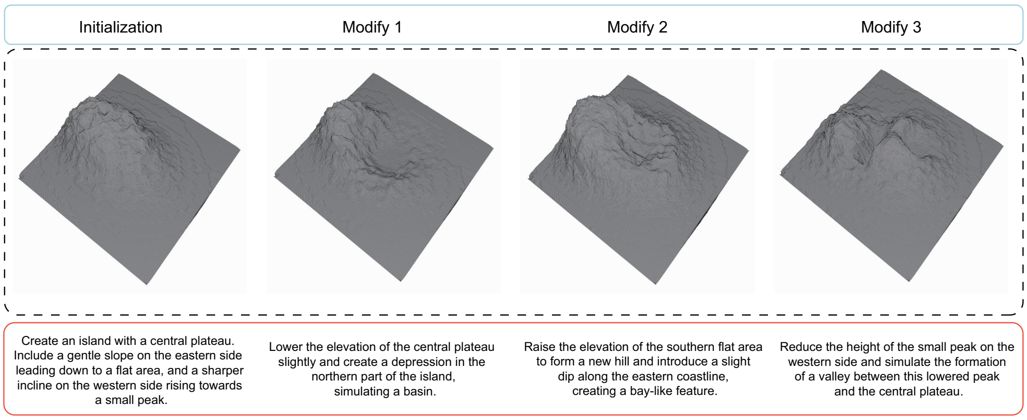

Initialization
Modify 1
Modify 2
Modify 3
Create an island with a central plateau. Include a gentle slope on the eastern side leading down to a flat area, and a sharper incline on the western side rising towards a small peak.
Lower the elevation of the central plateau slightly and create a depression in the northern part of the island, simulating a basin.
Raise the elevation of the southern flat area to form a new hill and introduce a slight dip along the eastern coastline, creating a bay-like feature.
Reduce the height of the small peak on the western side and simulate the formation of a valley between this lowered peak and the central plateau.

Figure 7. This image illustrates another significant role of the multiagent update strategy besides optimizing existing maps: continuously updating the terrain to simulate changes in the landscape over time.

The findings from this study were derived exclusively from experiments conducted on the hard dataset, chosen for its complexity to maximally accentuate the impact of each individual component. The outcomes, as detailed in the subsequent table, juxtapose the efficacy of our comprehensive approach against variations lacking in these critical elements.

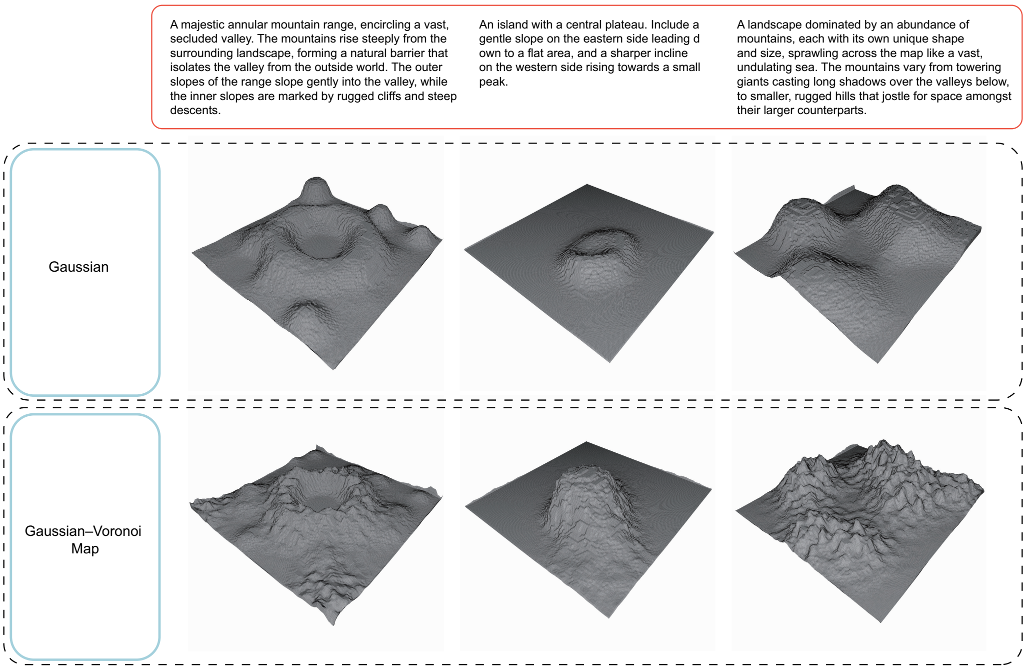
As shown in Table 5, the ablation study conducted on our text-to-terrain generation method underscores the indispensable role each component plays in crafting high-quality, realistic, and complex terrains from textual descriptions. By systematically removing key components and observing the impact on terrain generation, we have gained valuable insights into how each part contributes to the overall efficacy of the method. This analysis highlights not only the individual importance of these components but also their synergistic effect, which is crucial for achieving the best results.

Table 5. Ablation experiments result.

Configuration	Regular Score	Adjusted Score
Without Gaussian–Voronoi map	2.15	2.46
Without chain-of-thought behavior tree	2.58	2.60
Complete method	3.33	3.27

Due to the quality assessment experiments already reflecting the effectiveness and quality of the multiagent update strategy adequately, a separate ablation study for this component is not conducted.

The Gaussian–Voronoi map, in particular, stands out as a critical element for simulating the randomness and complexity inherent in natural terrains. This component effectively elevates low-frequency Gaussian data to high-frequency Voronoi data, capturing the intricate details and variations found in real-world landscapes. The absence of the Gaussian–Voronoi map in our experiments led to generated terrains that were notably less detailed and more uniform, lacking the natural variability and complexity that characterize genuine terrains. This demonstrates that the Gaussian–Voronoi map is essential for adding depth and realism to the generated terrains, making them more believable and engaging. Figure 8 illustrates the stark contrasts between terrains generated with and without the Gaussian–Voronoi map.



A majestic annular mountain range, encircling a vast, secluded valley. The mountains rise steeply from the surrounding landscape, forming a natural barrier that isolates the valley from the outside world. The outer slopes of the range slope gently into the valley, while the inner slopes are marked by rugged cliffs and steep descents.

An island with a central plateau. Include a gentle slope on the eastern side leading d own to a flat area, and a sharper incline on the western side rising towards a small peak.

A landscape dominated by an abundance of mountains, each with its own unique shape and size, sprawling across the map like a vast, undulating sea. The mountains vary from towering giants casting long shadows over the valleys below, to smaller, rugged hills that jostle for space amongst their larger counterparts.

Figure 8. In the first scenario, we directly converted the recognition results of the chain-of-thought behavior tree into depth maps. In the second scenario, we mapped these results onto the Gaussian–Voronoi map before converting them into depth maps. Utilizing the Gaussian–Voronoi map significantly boosts terrain diversity and accurately mimics the randomness found in natural landscapes. This approach not only enriches the terrain's visual appeal but also modifies elevation change rates, offering a more dynamic and realistic terrain modeling. Additionally, the Gaussian–Voronoi map also aids in altering the rate of elevation change across the terrain.

The CoT behavior tree significantly enhances the terrain generation process by accurately interpreting complex textual descriptions and converting them into detailed terrain features. It ensures logical consistency and contextual understanding throughout the text analysis, crucial for generating terrains that closely match the descriptions. It enables the large language model (LLM) to navigate through various layers of text input, extracting and interpreting spatial information and geographical concepts with precision. The chain-of-thought behavior tree guides the LLM in a structured manner, allowing it to maintain context and make logical inferences over multiple iterations of text analysis. Without this component, the terrain generation process becomes significantly less accurate in matching the textual descriptions to the generated 3D models. This loss of fidelity in interpretation underlines the chain-of-thought behavior tree's importance in ensuring that the generated terrains faithfully reflect the described landscapes, thereby enhancing the quality and relevance of the output. Figure 9 illustrates the CoT behavior tree's impact, demonstrating its ability to precisely identify and interpret comparative and positional information from text. This leads to a more accurate and contextually relevant terrain generation, as opposed to models that lack this structured analysis approach.

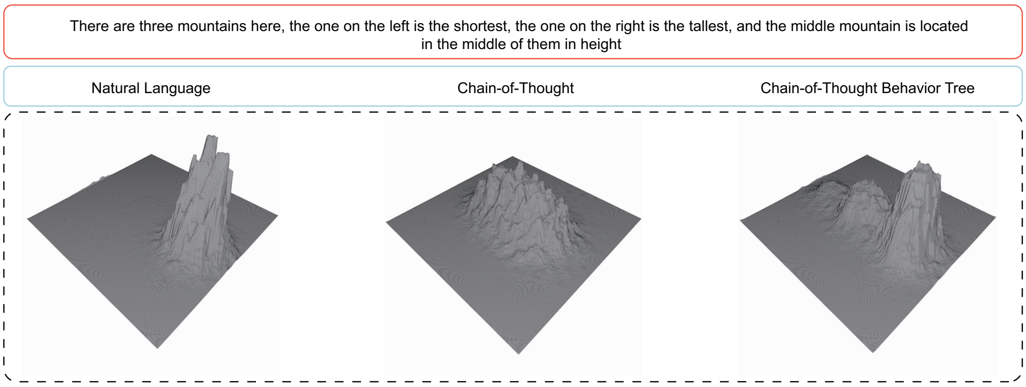

There are three mountains here, the one on the left is the shortest, the one on the right is the tallest, and the middle mountain is located in the middle of them in height
Natural Language
Chain-of-Thought
Chain-of-Thought Behavior Tree

Figure 9. We attempted to input a piece of text containing both comparative and positional information into the LLM. The LLM, which applied the chain-of-thought behavior tree, effectively identified this type of information and generated data that matched the description. On the other hand, using only the chain-of-thought-based LLM, although it recognized the three types of information, it did not arrange the positions of the three mountains well during data generation, causing them to blend together. In contrast, the natural-language-processing-based LLM was unable to fully recognize these contents. This result further demonstrates the effectiveness of the chain-of-thought behavior tree and its relative accuracy and stability in processing data compared with the chain-of-thought.

5. Discussion and Conclusions

From the perspective of model quality, our method provides the highest quality in the specialized task of generating 3D terrain models in the text-to-3D domain, significantly surpassing the capabilities of existing diffusion-model-based methods and offering highly flexible secondary editing capabilities for the models. This superior performance is primarily based on the three steps of our method that completely bypass diffusion models. First, we introduced the Gaussian–Voronoi map, which receives data in the form of Gaussians and maps them onto a map segmented by Voronoi, ensuring the method's ability to generate complex terrains from simple outputs. Then, we utilized the chain-of-thought behavior tree to extract consistent terrain information from diverse text sequences, ensuring that the method can extract terrain features from text sequences, and adopted a multiagent strategy to segment and process terrain features, achieving high-quality local updating of 3D terrains. Our research demonstrates that abandoning diffusion models and directly reading 3D data from text to generate 3D models is feasible and holds considerable promise, warranting further investigation.

However, our method also presents some issues that merit further research. Our method did not exhibit good robustness when faced with text inputs with significant metric scale errors. We believe that designing a more sophisticated chain-of-thought strategy will greatly contribute to addressing the aforementioned issues. Specifically, to improve the robustness of our method when faced with text inputs containing significant metric scale errors, we propose two approaches. First, we can preprocess the input text by comparing it with historical data to check for consistency in metric information. This preliminary step would help identify and rectify scale discrepancies before they affect the terrain generation process. On the other hand, we can enhance the model's understanding of metric data through fine-tuning. By preloading a set of feasible metric data and fine-tuning the large language model, we aim to achieve self-correction of metric errors. These strategies will contribute to a more accurate interpretation of spatial dimensions and enhance the overall robustness of our terrain synthesis method. In our forthcoming research, our goal is to explore methods for transferring Gaussians into three-dimensional space for representation, which will allow us to generate terrains with more complex features, such as caves and waterfalls, to our 3D terrain.

In terms of method speed, we believe that employing a strategy of dividing the map into sections and using a multithreaded approach to concurrently invoke multiple LLM processes for individual map modifications can significantly enhance the efficiency of our method, approaching real-time performance. Regarding computational performance, the primary limitation lies in the performance overhead of the large language models. The inherent overhead of our method is relatively low. Therefore, utilizing online large language model service APIs or switching to smaller large language models could greatly improve our efficiency.

In conclusion, our method innovatively employs a Gaussian–Voronoi map and a chain-of-thought behavior tree to facilitate the direct synthesis of 3D terrain from textual descriptions, successfully bypassing the limitations associated with traditional diffusion models. This approach not only demonstrates superior performance in generating complex 3D terrains with high accuracy but also provides a flexible framework for secondary terrain editing, enabling dynamic adjustments based on updated textual input. The introduction of multiagent strategies further enhances the local updating capabilities, ensuring that the generated terrains are both detailed and consistent with the provided text specifications. Our findings suggest that the direct interpretation of text into 3D data is not only viable but also highly efficient, offering a new perspective in the text-to-3D domain. The ability to generate detailed and controllable 3D terrains from text opens up new avenues for applications in virtual reality, gaming, and simulation environments. By advancing the state of the art in text-to-3D synthesis, our work paves the way for future research in the field, promising enhancements in both the quality of 3D model generation and the efficiency of the process, particularly in real-time and interactive applications.

Author Contributions: L.C. was responsible for the conceptualization, methodology, software development, data curation, writing—original draft preparation, and visualization. G.Z. contributed to validation, resources, writing—review and editing, and funding acquisition. Y.Z. carried out the formal analysis. Y.L. was in charge of the investigation. Y.G. supervised the project. X.C. managed the project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Project of Humanities and Social Sciences of the Ministry of Education in China, grant number 22YJAZH002.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Due to laboratory confidentiality agreements and considerations for participant privacy, we are unable to publicly disclose our experimental data. However, we welcome inquiries from researchers, including anonymous reviewers, who wish to access the complete dataset. Such requests should be directed to the second author of this paper, after stating the necessary reasons for data access. Regarding the project source code, it will be made available on our project homepage link after the end of the project cycle and upon acceptance and publication of the paper. The dataset used in the project has already been made available at the following URL: https://github.com/

FI3GS/Zero-Shot-Text-to-3D-Terrain-Synthesis-Using-Only-Large-Language-Model, accessed on 9 March 2024.

Acknowledgments: We would like to express our gratitude to the pioneers in the field, the various research achievements cited in this paper. We extend our thanks to our family members for their unwavering support and encouragement. To the successors, your future citations and references to this work are crucial for broadening the impact of our research and extending its life. We are also grateful to our peers, the companions of our academic journey, whose shared efforts and struggles in learning and striving have left an indelible mark on our hearts.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CoT	chain-of-thought
ToT	tree-of-thought
LLM	large language model

References

1. Xu, J.; Wang, X.; Cheng, W.; Cao, Y.-P.; Shan, Y.; Qie, X.; Gao, S. Dream3D: Zero-Shot Text-to-3D Synthesis Using 3D Shape Prior and Text-to-Image Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 20908–20918.
2. Croitoru, F.A.; Hondru, V.; Ionescu, R.T.; Shah, M. Diffusion Models in Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 10850–10869. [CrossRef] [PubMed]
3. Kweon, I.; Kanade, T. Extracting Topographic Terrain Features from Elevation Maps. *CVGIP: Image Underst.* **1994**, *59*, 171–182. [CrossRef]
4. Sultan, F.; Farley, J.U.; Lehmann, D.R. A Meta-Analysis of Applications of Diffusion Models. *J. Mark. Res.* **1990**, *27*, 70–77. [CrossRef]
5. Aboulaich, R.; Meskine, D.; Souissi, A. New Diffusion Models in Image Processing. *Comput. Math. Appl.* **2008**, *56*, 874–882. [CrossRef]
6. Li, X.; Liu, Y.; Lian, L.; Yang, H.; Dong, Z.; Kang, D.; Zhang, S.; Keutzer, K. Q-Diffusion: Quantizing Diffusion Models. In Proceedings of the IEEE/CVF International Conference on Computer Vision, France, Paris, 2–6 October 2023.
7. Chen, Z.; Zhang, H. Learning Implicit Fields for Generative Shape Modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
8. Luo, A.; Li, T.; Zhang, W.-H.; Lee, T.S. SurfGen: Adversarial 3D Shape Synthesis with Explicit Surface Discriminators. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 16238–16248.
9. Wu, R.; Zheng, C. Learning to Generate 3D Shapes from a Single Example. *arXiv* **2022**, arXiv:2208.02946.
10. Wu, R.; Zhuang, Y.; Xu, K.; Zhang, H.; Chen, B. PQ-Net: A Generative Part Seq2Seq Network for 3D Shapes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
11. Zheng, X.-Y.; Liu, Y.; Wang, P.-S.; Tong, X. SDF-StyleGAN: Implicit SDF-Based StyleGAN for 3D Shape Generation. *Comput. Graph. Forum (SGP)* **2022**, *41*, 52–63. [CrossRef]
12. Cao, Y.P.; Liu, Z.N.; Kuang, Z.F.; Kobbelt, L.; Hu, S.M. Learning to Reconstruct High-Quality 3D Shapes with Cascaded Fully Convolutional Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2018; pp. 616–633.
13. Li, J.; Xu, K.; Chaudhuri, S.; Yumer, E.; Zhang, H.; Guibas, L. GRASS: Generative Recursive Autoencoders for Shape Structures. *ACM Trans. Graph. (TOG)* **2017**, *36*, 1–14. [CrossRef]
14. Tatarchenko, M.; Dosovitskiy, A.; Brox, T. Octree Generating Networks: Efficient Convolutional Architectures for High-Resolution 3D Outputs. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
15. Cai, R.; Yang, G.; Averbuch-Elor, H.; Hao, Z.; Belongie, S.; Snavely, N.; Hariharan, B. Learning Gradient Fields for Shape Generation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 364–381.
16. Xiang, P.; Wen, X.; Liu, Y.S.; Cao, Y.P.; Wan, P.; Zheng, W.; Han, Z. Snowflake Point Deconvolution for Point Cloud Completion and Generation with Skip-Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 6320–6338. [CrossRef]
17. Yang, G.; Huang, X.; Hao, Z.; Liu, M.Y.; Belongie, S.; Hariharan, B. PointFlow: 3D Point Cloud Generation with Continuous Normalizing Flows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.

18. Vahdat, A.; Williams, F.; Gojcic, Z.; Litany, O.; Fidler, S.; Kreis, K. LION: Latent Point Diffusion Models for 3D Shape Generation. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2022**, *35*, 10021–10039.
19. Gao, L.; Wu, T.; Yuan, Y.J.; Lin, M.X.; Lai, Y.K.; Zhang, H. TM-Net: Deep Generative Networks for Textured Meshes. *ACM Trans. Graph. (TOG)* **2021**, *40*, 263:1–263:15. [CrossRef]
20. Gao, L.; Yang, J.; Wu, T.; Yuan, Y.J.; Fu, H.; Lai, Y.K.; Zhang, H. SDM-Net: Deep Generative Network for Structured Deformable Mesh. *ACM Trans. Graph. (TOG)* **2019**, *38*, 1–15. [CrossRef]
21. Gupta, K. Chandraker, Manmohan. Neural Mesh Flow: 3D Manifold Mesh Generation via Diffeomorphic Flows. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1747–1758.
22. Tang, J.; Ren, J.; Zhou, H.; Liu, Z.; Zeng, G. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. *arXiv* **2023**, arXiv:2309.16653.
23. Yi, T.; Fang, J.; Wu, G.; Xie, L.; Zhang, X.; Liu, W.; Tian, Q.; Wang, X. GaussianDreamer: Fast Generation from Text to 3D Gaussian Splatting with Point Cloud Priors. *arXiv* **2023**, arXiv:2310.08529.
24. Ren, J.; Pan, L.; Tang, J.; Zhang, C.; Cao, A.; Zeng, G.; Liu, Z. DreamGaussian4D: Generative 4D Gaussian Splatting. *arXiv* **2023**, arXiv:2312.17142.
25. Müller, T.; Evans, A.; Schied, C.; Keller, A. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph. (ToG)* **2022**, *41*, 1–15. [CrossRef]
26. Kerbl, B.; Kopanas, G.; Leimkühler, T.; Drettakis, G. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* **2023**, *42*, 1–14. [CrossRef]
27. Jain, A.; Mildenhall, B.; Barron, J.T.; Abbeel, P.; Poole, B. Zero-Shot Text-Guided Object Generation with Dream Fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 867–876.
28. Lee, H.-H.; Chang, A.X. Understanding Pure CLIP Guidance for Voxel Grid NeRF Models. *arXiv* **2022**, arXiv:2209.15172.
29. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. REACT: Synergizing Reasoning and Acting in Language Models. *arXiv* **2022**, arXiv:2210.03629.
30. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. Self-Refine: Iterative Refinement with Self-Feedback. *arXiv* **2023**, arXiv:2303.17651.
31. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le Q.V.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
32. Jin, F.; Liu, Y.; Tan, Y. Zero-Shot Chain-of-Thought Reasoning Guided by Evolutionary Algorithms in Large Language Models. *arXiv* **2024**, arXiv:2402.05376.
33. Fatemi, B.; Halcrow, J.; Perozzi, B. Talk Like a Graph: Encoding Graphs for Large Language Models. *arXiv* **2023**, arXiv:2310.04560.
34. Perozzi, B.; Fatemi, B.; Zelle, D.; Tsitsulin, A.; Kazemi, M.; Al-Rfou, R.; Halcrow, J. Let Your Graph Do the Talking: Encoding Structured Data for LLMs. *arXiv* **2024**, arXiv:2402.05862.
35. Guo, J.; Du, L.; Liu, H. GPT4Graph: Can Large Language Models Understand Graph Structured Data? An Empirical Evaluation and Benchmarking. *arXiv* **2023**, arXiv:2305.15066.
36. Chai, Z.; Zhang, T.; Wu, L.; Han, K.; Hu, X.; Huang, X.; Yang, Y. GraphLLM: Boosting Graph Reasoning Ability of Large Language Model. *arXiv* **2023**, arXiv:2310.05845.
37. Ge, Y.; Liu, S.; Feng, W.; Mei, L.; Chen, L.; Cheng, X. Graph Descriptive Order Improves Reasoning with Large Language Model. *arXiv* **2024**, arXiv:2402.07140.
38. Liu, R.; Yang, R.; Jia, C.; Zhang, G.; Yang, D.; Vosoughi, S. Training Socially Aligned Language Models on Simulated Social Interactions. In Proceedings of the The Twelfth International Conference on Learning Representations, Vienna, Austria, 7 May 2023.
39. Liu, Z.; Zhang, Y.; Li, P.; Liu, Y.; Yang, D. Dynamic LLM-Agent Network: An LLM-Agent Collaboration Framework with Agent Team Optimization. *arXiv* **2023**, arXiv:2310.02170.
40. Dubois, Y.; Li, C.X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P.S.; Hashimoto, T.B. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. *Adv. Neural Inf. Process. Syst.* **2024**, *36*.
41. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; Narasimhan, K. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *arXiv* **2024**, arXiv:2305.10601.
42. Park, J.S.; O'Brien, J.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative Agents: Interactive Simulacra of Human Behavior. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, San Francisco, CA, USA, 29 October–1 November 2023.
43. Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; Yao, S. Reflexion: Language Agents with Verbal Reinforcement Learning. *arXiv* **2024**, arXiv:2303.11366.
44. Xu, Z.; Yu, C.; Fang, F.; Wang, Y.; Wu, Y. Language Agents with Reinforcement Learning for Strategic Play in the Werewolf Game. *arXiv* **2023**, arXiv:2310.18940.
45. Huang, Z.; Wu, T.; Jiang, Y.; Chan, K.C.; Liu, Z. ReVersion: Diffusion-Based Relation Inversion from Images. *arXiv* **2023**, arXiv:2303.13495.

46. Guédon, A.; Lepetit, V. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *arXiv* **2023**, arXiv:2311.12775.
47. Tang, J.; Zhou, H.; Chen, X.; Hu, T.; Ding, E.; Wang, J.; Zeng, G. Delicate textured mesh recovery from nerf via adaptive surface refinement. *arXiv* **2023**, arXiv:2303.02091.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.