T^{3}Bench: Benchmarking Current Progress in Text-to-3D Generation

Yuze He[1]*, Yushi Bai[1]*, Matthieu Lin[1], Wang Zhao[1], Yubin Hu[1], Jenny Sheng[1], Ran Yi[2], Juanzi Li[1], Yong-Jin Liu[1]✉

[1]Tsinghua University [2]Shanghai Jiao Tong University

{hyz22,bys22}@mails.tsinghua.edu.cn, liuyongjin@tsinghua.edu.cn

Abstract

Recent methods in text-to-3D leverage powerful pre-trained diffusion models to optimize NeRF. Notably, these methods are able to produce high-quality 3D scenes without training on 3D data. Due to the open-ended nature of the task, most studies evaluate their results with subjective case studies and user experiments, thereby presenting a challenge in quantitatively addressing the question: How has current progress in Text-to-3D gone so far? In this paper, we introduce T^{3}Bench, the first comprehensive text-to-3D benchmark containing diverse text prompts of three increasing complexity levels that are specially designed for 3D generation. To assess both the subjective quality and the text alignment, we propose two automatic metrics based on multi-view images produced by the 3D contents. The quality metric combines multi-view text-image scores and regional convolution to detect quality and view inconsistency. The alignment metric uses multi-view captioning and Large Language Model (LLM) evaluation to measure text-3D consistency. Both metrics closely correlate with different dimensions of human judgments, providing a paradigm for efficiently evaluating text-to-3D models. The benchmarking results, shown in Fig. 1, reveal performance differences among six prevalent text-to-3D methods. Our analysis further highlights the common struggles for current methods on generating surroundings and multi-object scenes, as well as the bottleneck of leveraging 2D guidance for 3D generation. Our project page is available at: https://t3bench.com.

1. Introduction

> *It is a narrow mind which cannot look at a subject from various points of view.* — *George Eliot*

Equipping machines with the ability to automatically generate 3D objects and scenes from text descriptions has long been an ambitious and ongoing pursuit. Recent methods, such as diffusion model [9, 24] and NeRF [7, 17, 37], have significantly improved the effectiveness of text-to-3D methods, empowering potential applications ranging from arts realization to industrial design.

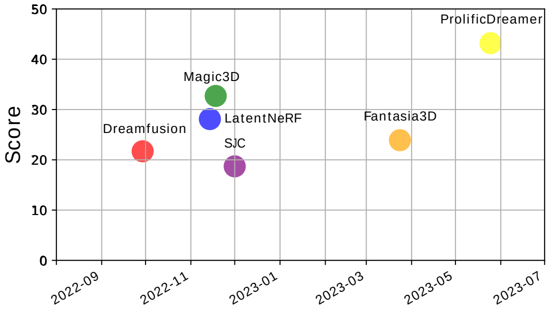

Figure 1. The average scores of six prevalent text-to-3D methods on T^{3}Bench, computed by the mean of quality & alignment metrics.

However, there lacks a systematic approach in benchmarking current progress on text-to-3D methods, which is most prominently reflected in two aspects: (a) A lack of a standard set of diverse, challenging test textual inputs. (b) An absence of a set of automatic and comprehensive evaluation metrics to quantitatively measure the quality of the generated 3D scenes. Specifically, previous works [12, 29, 30] mostly adopt simple object or scene prompts for evaluation, and largely rely on subjective user experiments. Several works [18, 21, 32, 36] assess 3D generation quality by rendering the generated 3D model into a single 2D image and measuring its alignment with the text prompt through CLIP cosine distance or CLIP R-precision. Nevertheless, they only consider **one view** of the 3D scene, failing to assess the overall 3D quality.

To facilitate further research in this direction, we introduce the first comprehensive text-to-3D benchmark, namely T^{3}Bench. For a careful and thorough assessment, we build

[*] Equal contribution

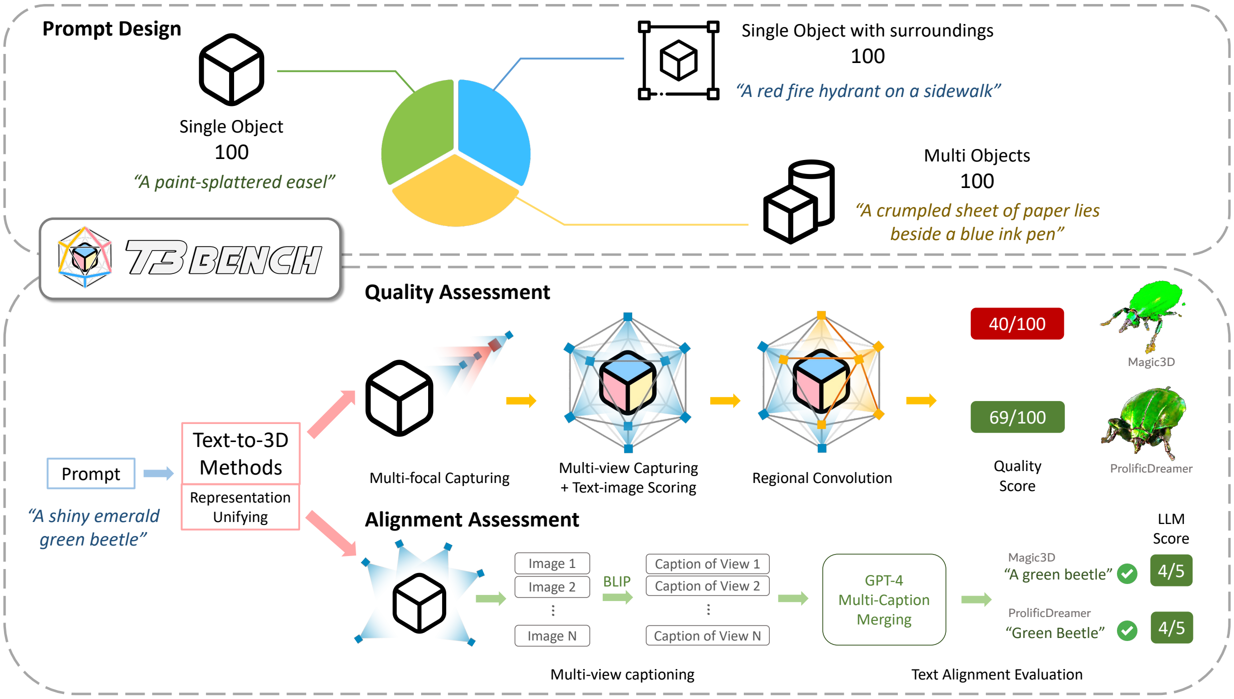

Figure 2. The overview of our T^{3}Bench benchmark pipeline.

the benchmark such that it can accurately reflect the primary challenges of current text-to-3D approaches. This includes their scalability and robustness in generating a variety of 3D scenes, the quality and view consistency of these generated scenes, and the correctness or alignment of these 3D scenes with their respective texts. Specifically, we devise three prompt suites incorporating diverse 3D scenes and with increasing complexity, including *Single object*, *Single object with surroundings*, and *Multiple objects*. We also propose two automatic evaluation metrics that both take **multi-view information** into consideration, with each focusing on assessing the subjective quality of the generated 3D scenes and its alignment with the textual prompt respectively. To calculate these two metrics, we first employ multi-focal and multi-view capturing to obtain a set of 2D images from the generated 3D scenes. The *quality* metric individually scores these multi-view images with text-image scoring models (CLIP [23], ImageReward [35]), and then combines them into one overall quality measurement using regional convolution, which also effectively detects the infamous Janus problem (view inconsistency) in prevalent text-to-3D models [10, 21]. On the other hand, the *alignment* metric utilizes multi-view captioning and LLM (large language model) evaluation to measure how closely the 3D information aligns with the textual information in the input text prompt. Our user experiments show that both metrics correlate closely with human scorings in 1-5 (with a Spearman correlation higher than 0.75), which supports that these two metrics are efficient and automatic measurements.

As the first attempt to benchmark current text-to-3D methods, T^{3}Bench yields fruitful results. Our benchmark reveals the strengths and weaknesses across six prevalent text-to-3D methods, as well as their common insufficiency when faced with more complicated 3D scenes, such as those involving multiple objects. We also analyze the correlation between the performance of text-to-3D methods and the quality of the 2D guidance generated by diffusion models, showing that the primary hurdle for text-to-3D mainly lies in the transition from 2D to a consistent 3D scene.

2. Related Works

Text-to-3D. Predominant works in text-to-3D [4, 12, 16, 20, 22, 34] circumvent the need for 3D training data by using large pretrained text-to-image diffusion models [25, 27]. However, these approaches suffer from inconsistency between views. Notably, the proposed score distillation loss [20] does not take into account the consistency between views as the diffusion model mimics a stochastic process [9]. On the one hand, ProlificDreamer [34] proposes a variational formulation of the score distillation loss to consider the stochasticity in the diffusion process. On the other hand, some researches propose to fine-tune the diffusion to improve its consistency across views [39]. These current metrics do not adequately consider the 3D nature of the generated results, which makes it difficult to compare the effectiveness of different methods. Prior work has relied either on labor-intensive user studies [34] or CLIP R-precision [23], which does not consider 3D consistency. While early attempts have been made to measure 3D consistency [10], these efforts only capture one aspect of the problem and overlook crucial metrics such as quality and prompt alignment.

Text-to-image Generation and Evaluation. With the development of diffusion models [9], text-based image generation has experienced significant progress in recent years [25, 27]. These models excel at complex tasks like editing and composition [3, 8]. However, comparing their capabilities in text-based generation is challenging due to the open-ended nature of the task [2]. Prior work in text-to-image generation introduced DrawBench [27], a comprehensive set of prompts aiming to evaluate various aspects, including color understanding, object recognition, and spatial relations. Other approaches leverage CLIP [23] and BLIP [11] to measure the similarity between text and generated images by using these models as scorers to gauge prompt alignment. In a similar vein, the Aesthetic score [28] employs the CLIP model to predict image aesthetics. While these methods assess alignment and quality to some extent, they fall short in considering multiple properties like toxicity, quality, and alignment. To encompass these diverse properties into a single model, ImageReward [35] proposes training a reward model via reinforcement learning from human feedback. Results show that this reward model better aligns with human preferences. Although evaluation for 3D generation can draw on text-to-image evaluation methods, it is important to note the major difference between the two: 3D contains semantic information from multiple viewpoints rather than a single view.

3. Method

In this section, we discuss the methodology in constructing T^{3}Bench, including the design and generation of text prompts, the unification of 3D representations, and the introduction of two novel evaluation metrics — the quality assessment and the alignment assessment.

3.1. Prompt Design

While there are some widely used text-to-image prompt sets, such as DrawBench [26] and DALL-EVAL [5], many of the prompts in these benchmarks pose substantial challenges for existing text-to-3D methods and lack an adequate degree of distinction. Certain prompts, for instance, are excessively lengthy, while others incorporate complex aspects such as counting, leading to poor 3D scenes generated by all current text-to-3D methods. Therefore, a new set of prompts needs to be specifically crafted for evaluating prevalent text-to-3D methods.

We observe that current text-to-3D approaches demonstrate relatively robust performance on prompts with a single object. However, their performance notably declines on text prompts that include environmental surroundings or multiple objects. Such deficiency is partly due to the utilization of 2D supervisions, which cannot ensure consistency amongst different viewpoints. With these observations, we design three prompt sets with increasing complexity to perform a targeted evaluation of text-to-3D approaches, namely *Single object*, *Single object with surroundings*, and *Multiple objects*. The *Single object* set represents the simplest scenario to establish a baseline level of performance, and the other two prompt sets introduce increased levels of difficulty by incorporating additional information like surroundings or multiple objects.

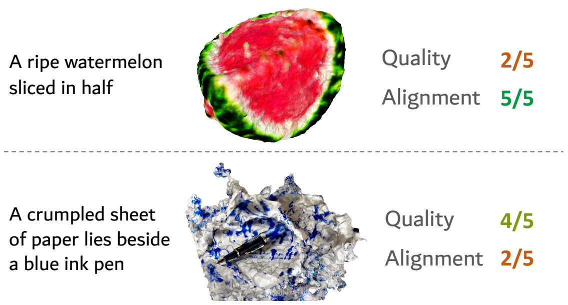
A ripe watermelon sliced in half
Quality 2/5
Alignment 5/5
A crumpled sheet of paper lies beside a blue ink pen
Quality 4/5
Alignment 2/5

Figure 3. The subjective 3D quality and the alignment with the text prompt are both vital to evaluate a 3D generation result.

To generate these prompt sets, we first use GPT-4 [19] to generate a large pool of candidate prompts, and then manually filter out prompts that contain proper nouns or toponyms. Subsequently, we utilize ROUGE-L [13] to quantify prompt similarity and incrementally remove highly similar prompts until there remains a number of N distinct prompts with significant diversity in each prompt set.

3.2. Unified 3D Representation

Different text-to-3D methods may employ various 3D representations during generation, such as NeRF [17] and 3D mesh. From a testing perspective, a 3D mesh is more conducive than NeRF due to its explicit geometric structure, which facilitates localization and normalization. Moreover, the primary use of text-to-3D is to obtain editable 3D assets that can be applied in fields such as virtual reality and gaming. Considering the purpose and practical applications, 3D mesh is a more suitable unified representation for benchmarking text-to-3D methods. The NeRF generated by text-to-3D methods can be converted into a 3D mesh using either DMTet [31] or Marching Cube [14] algorithm, and we choose the one that produces superior results. This makes subsequent evaluations more convenient while encouraging the generation of 3D scenes with more compact and clear geometry.

3.3. Evaluation Metrics

3.3.1 Overview

The evaluation of text-to-3D methods remains challenging due to the need to fully account for the quality, view consistency, and text alignment of the generated 3D scenes.

Our evaluation metrics primarily focus on two dimensions that typically reflect the effectiveness of text-to-3D methods: the subjective quality of the generated 3D scene, and the degree of alignment between the generated 3D scene and the input text prompt. The example in Fig. 3 shows that both dimensions are crucial, as some methods may accurately generate objects described in the prompt but lack subjective quality (top), while others may produce high-quality objects yet deviate from the text or fail to encapsulate all the information described in the text prompt (bottom).

To assess quality, we devise a scoring mechanism that comprises multi-focal and multi-view capturing, and utilizes text-image scoring models to obtain an overall quality measurement of the generated 3D scene. As for the textual alignment, we develop a scoring metric based on multi-view captioning and LLM evaluation. We offer a comprehensive explanation of these metrics in the following sections.

3.3.2 Quality Assessment

Since the spatial geometry information is crucial for the generated 3D scenes, evaluation from a single view is inadequate to assess the quality of the generated results. We believe a comprehensive and reliable 3D quality assessment should take into account the following issues: (a) *Viewpoint selection*: Choosing an appropriate viewpoint can better reflect the quality of the 3D scene, particularly potential object occlusions; (b) *Area coverage*: It is essential to simultaneously examine the current viewpoint and adjacent areas. By doing so, the assessment can take into account a more global geometry, thereby avoiding a collapse to a local optimal view that leads to failure in detecting 3D consistency issues like the Janus problem.

To meet these conditions, we incorporate a delicate capturing and scoring procedure to evaluate the quality of the 3D generation. The following steps outline our method.

Mesh Normalization. We convert the generated 3D scene into a mesh and scale it proportionally in the x, y, and z directions, allowing the mesh to fit within a cube with a range of $[-1, 1]$ on all three dimensions. This helps to roughly determine the mesh's range for subsequent capturing.

Multi-Focal Capturing. Capturing a 2D image from a single location using a fixed focal length may yield inaccurate evaluation results. This is because the information in the captured image may be incomplete when the focal length is too long, and may occupy only a small portion in the frame when the focal length is too short. To address this issue, we employ five different focal lengths to capture the mesh at each location and select the best focal length based on the highest text-image score.

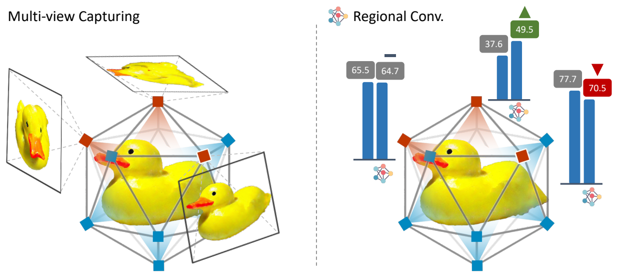

Figure 4. Demonstration of scores at different viewpoints after multi-view capturing and regional convolution. Here we use a level-0 icosahedron for a schematic illustration, please refer to Fig. 7 for more details.

Multi-View Capturing. To capture the 3D scene as completely as possible, we construct an icosahedron with a radius of 2.2 around the origin and capture the 3D scene from all the vertices of the icosahedron (see an illustration of icosahedron in Fig. 4,7). As text-image scoring models may be sensitive to rotation, we ensure that the plane formed by the up vector and look-at vector during capture contains the vertical axis. In practice, we use a level-2 icosahedron and capture from 161 locations.

Scoring and Regional Convolution. We employ text-image scoring models, such as CLIP [23] and ImageReward [35], to score the 2D views from all 161 icosahedron vertices along with the textual prompt. To capture a more global feature, we consider applying a pooling operator to the score at each location. Standard averaging of scores across all locations may not be appropriate, as most views, e.g., top or bottom, are not suitable for evaluation, and this approach may oversmooth the actual performance. Meanwhile, taking the overall maximum scores may overlook the view inconsistency issue. Therefore, we design a regional convolution mechanism to smooth out the score over each local region. We treat the icosahedron as a graph composed of vertices and edges, and perform mean pooling on the graph with the following recursive formula:

$$s_i^{(t+1)} = \frac{1}{|N(i)|+1} \left(s_i^{(t)} + \sum_{j \in N(i)} s_j^{(t)} \right), \qquad (1)$$

where $s_i^{(t)}$ is the score of point i on the icosahedron at the t-th iteration, $N(i)$ is the set of neighboring points of i, and $|N(i)|$ is the number of neighbors of i. The superscript $(t+1)$ denotes the score after the $(t+1)$-th iteration. We choose a total of $t = 3$ iterations of mean pooling as we empirically find that it ensures a balance between adequate smoothing and over-smoothing.

After these steps, we select the highest score from all viewpoints as the final quality score for the 3D generation.

3.3.3 Alignment Assessment

In addition to the evaluation from the quality aspect, the alignment between 3D semantic information and text is another crucial aspect that should be considered. To measure the alignment between different modalities, we first perform 3D-to-text to caption the 3D scene and then compute the similarity between the caption and the textual prompt.

Directly utilizing image captioning methods such as BLIP [11] on a single view may fail to reflect the comprehensive information of a 3D object. To this end, we utilize a 3D-to-text caption pipeline similar to Cap3D [15]. Initially, a level-0 icosahedron consisting of 12 vertices is established around the origin. This icosahedron captures the 3D scene on the 12 locations, each of which is captioned using BLIP. We then employ GPT-4 [19] to merge these captions (detailed in Sec. 7.2), resulting in a 3D caption for the object.

Upon obtaining the 3D caption, we need to measure its alignment with the original prompt, particularly concerning the *recall* of the original prompt within the caption. Specifically, we observe that the text-to-3D methods might generate features not mentioned in the prompt (e.g., a red beak feature on a rubber duck), which may be reflected in the caption provided by BLIP. Such additional features should not be considered misalignments, even though many similarity-based scoring methods (BLEU, BERTScore [38]) might assign them lower scores. To assess the text recall, we adopt ROUGE-L [13]. We also incorporate large language models (LLMs) as text recall evaluators, drawing upon their demonstrated ability to mimic human experts in data annotation and evaluation [1]. Here is the prompt we use:

Prompt: You are an assessment expert responsible for prompt-prediction pairs. Your task is to score the prediction according to the following requirements:
1. Evaluate the recall, or how well the prediction covers the information in the prompt. If the prediction contains information that does not appear in the prompt, it should not be considered as bad.
2. If the prediction contains correct information about color or features in the prompt, you should also consider raising your score.
3. Assign a score between 1 and 5, with 5 being the highest. Do not provide a complete answer; give the score in the format: 3
Prompt: A photographer is capturing a beautiful butterfly with his camera
Prediction: A man photographing a butterfly near a tree and map, surrounded by plants
Answer: 4

4. Experiments

4.1. Metric Evaluation

In order to validate the reliability of our proposed metrics, we conduct a human-centered evaluation. Expert evaluators are tasked with manually assigning scores to 3D scenes generated by 6 different methods on 30% of all the prompts in T^{3}Bench. This results in a total of 1,080 scores. The human annotations span two dimensions: quality, which concerns the subjective quality of the generated results, and alignment, which focuses on the extent to which the generated content covers the original prompt. These evaluations are quantified using a 1-5 Likert scale. Subsequently, we measure the correlation between the automatic metrics and human annotations using Spearman's ρ, Kendall's τ, and Pearson's ρ correlation coefficients.

We compare the following set of metrics. For the quality metric, we compare the use of a single front view to our multi-view approach, as well as the employment of CLIP and ImageReward as text-image scoring models. For the alignment metric, we explore the use of ROUGE-L and LLM (GPT-4) to measure text recall. We report the results in Tab. 1. The first four columns reveal that multi-view capturing is superior to single-view examination. Moreover, compared to ROUGE-L, GPT-4 provides a more reliable assessment of alignment, as depicted by the last two columns. These findings justify the design of our processing and scoring methods in Sec. 3.3. Overall, we observe that Multi-view capturing + ImageReward and 3D captioning + GPT-4 scoring align most closely with quality and alignment aspects as annotated by human experts, respectively. We thus employ these combinations as the default quality and alignment metrics in our benchmark.

4.2. Benchmarking Results

Experimental Setup.[1] Following the prompt generation scheme outlined in Sec. 3.1 and taking into consideration both experimental breadth and test speed, we utilize GPT-4 to generate $N = 100$ prompts for each of the three categories: *single object*, *single object with surroundings*, and *multiple objects*, resulting in a total of 300 prompts. We employ the implementation provided by ThreeStudio [6] to uniformly evaluate six prevalent text-to-3D methods on these prompts, including DreamFusion [21], Magic3D [12], LatentNeRF [16], Fantasia3D [4], SJC [33], and ProlificDreamer [34]. We normalize the original scores on quality

[1] The data and evaluation code are available at https://github.com/THU-LYJ-Lab/T3Bench.

	Single-CLIP	Single-ImageReward	Multi-CLIP	Multi-ImageReward	3D Caption + ROUGE-L	3D Caption + GPT-4
Quality						
Spearman (ρ)	0.647	0.684	0.730	**0.752**	0.381	0.585
Kendall (τ)	0.505	0.539	0.584	**0.605**	0.293	0.505
Pearson (ρ)	0.629	0.664	0.707	**0.750**	0.367	0.537
Alignment						
Spearman (ρ)	0.648	0.628	0.740	0.701	0.537	**0.765**
Kendall (τ)	0.505	0.488	0.587	0.554	0.418	**0.690**
Pearson (ρ)	0.640	0.625	0.722	0.696	0.558	**0.761**

Table 1. The correlation of different combinations of evaluation methods with human annotations. Single-X and Multi-X correspond to the measurement X taken on single-view and multi-view of the generated 3D scenes.

and alignment assessment from the range $[-2.5, 2.5]$, $[1, 5]$ to $[0, 100]$. We set the five focal lengths used for multi-focal capturing to 3.0, 4.0, 5.0, 6.0, 7.5, and set the resolution of the rendered image to 512×512. When capturing the 3D mesh, we directly use the diffuse color of the texture at the corresponding direction as the rendering result, without additional usage of any light source.

To obtain optimal mesh extraction, DMTet is utilized for SJC, Magic3D, and Fantasia3D, while other methods employ the Marching Cube algorithm; then, to retain quality without excessive UV unwrapping times, textures are extracted following mesh geometry simplification to a maximum of 40,000 faces. For methods that yield 3D scenes with a diffusion latent radiance field representation rather than RGB, we also convert them into a latent texture map. Subsequently, we transform these into RGB textures using a latent decoder with a sliding window strategy to achieve anti-aliasing conversion.

Tab. 2 reports the quality, alignment, and the average scores for each text-to-3D method on the three prompt sets in T^{3}Bench. We also showcase some examples in Fig. 5. (More can be found in Sec. 9).

Comparison of different methods. We found Dreamfusion is limited by the resolution of diffusion guidance, where the textures of the generated objects are relatively simple, and it is challenging to generate complex geometry. This results in a relatively low score for Dreamfusion, particularly in the *Single Object set* where other methods have an advantage.

Both Magic3D and LatentNeRF outperform Dreamfusion, which can be attributed to their coarse-to-refine strategy that allows for higher resolution optimization of textures. This strategy enables better restoration of details in the text, resulting in improved quality and text alignment. However, neither of these two methods demonstrates significant advantages in modeling surroundings and multiple objects, as suggested by a noticeable decline in quality in the latter two prompt sets.

	Quality	Alignment	Average
Single Object			
Dreamfusion	24.9	24.0	24.4
Magic3D	38.7	35.3	37.0
LatentNeRF	34.2	32.0	33.1
Fantasia3D	29.2	23.5	26.4
SJC	26.3	23.0	24.7
ProlificDreamer	51.1	47.8	49.4
Single Object with Surroundings			
Dreamfusion	19.3	29.8	24.6
Magic3D	29.8	41.0	35.4
LatentNeRF	23.7	37.5	30.6
Fantasia3D	21.9	32.0	27.0
SJC	17.3	22.3	19.8
ProlificDreamer	42.5	47.0	44.8
Multiple Objects			
Dreamfusion	17.3	14.8	16.1
Magic3D	26.6	24.8	25.7
LatentNeRF	21.7	19.5	20.6
Fantasia3D	22.7	14.3	18.5
SJC	17.7	5.8	11.7
ProlificDreamer	45.7	25.8	35.8

Table 2. The average scores of text-to-3D methods on T^{3}Bench.

SJC, on the other hand, generates a large amount of floating density, making it difficult to extract high-quality 3D mesh, especially in complex scenes. This reduces its practical applicability, which is reflected in our metrics. Fantasia3D generates rich textures and achieves satisfactory single-object results, but its performance drops in complex scenes due to relatively imprecise geometry generation.

With the introduced Variational Score Distillation, ProlificDreamer optimizes the distribution of the scene and demonstrates clear advantages over other methods in both simple single-object scenes and more complex scenarios. However, the use of VSD sometimes introduces excessive irrelevant information or geometry noise, which may have a negative impact on human perception and BLIP captioning. As the number of objects increases, this leads to a decrease in the alignment metric advantages.

Trends across different prompt sets. As shown in Tab. 2, for the *Single Object* set, the overall performance is rela-

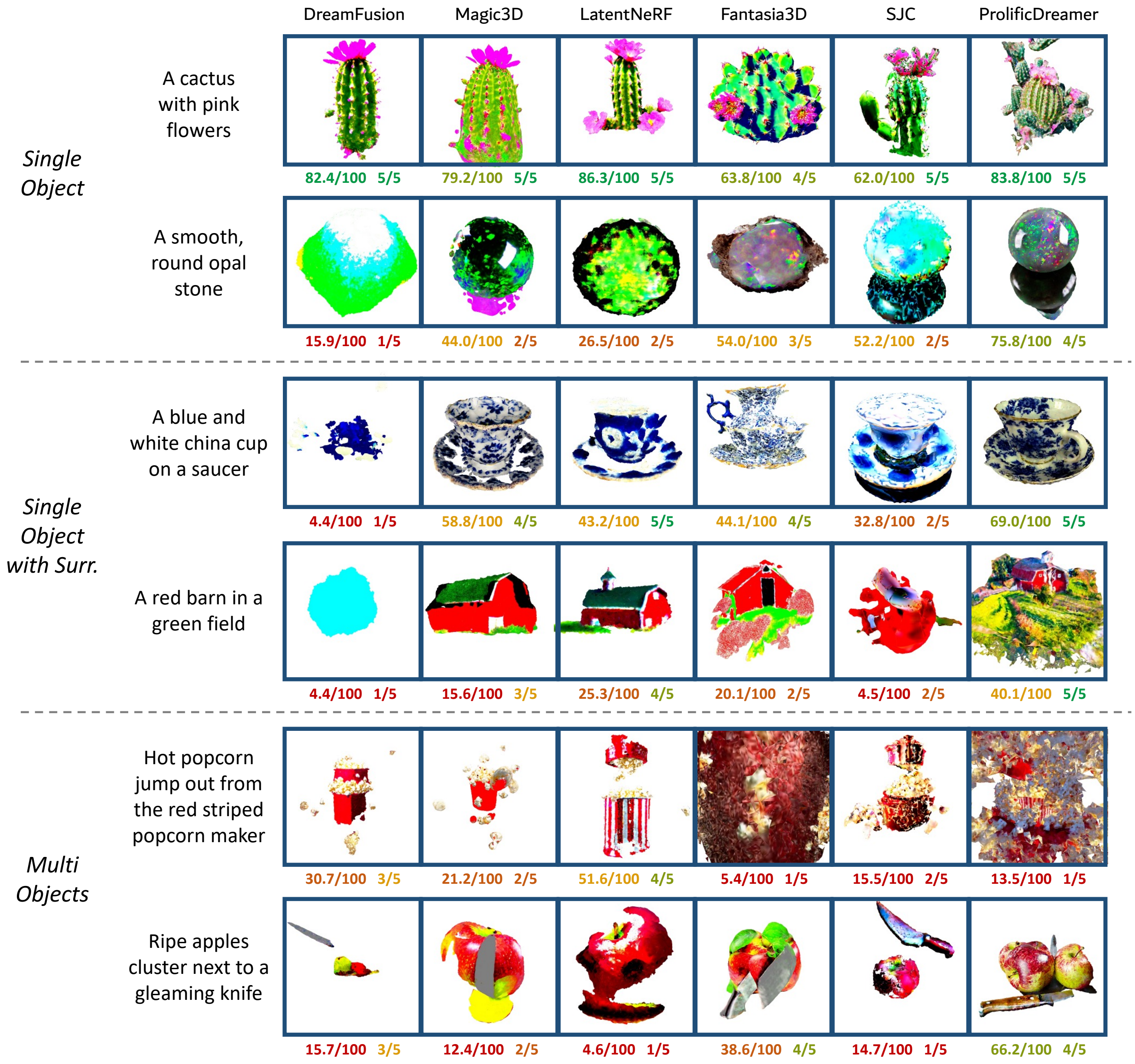

Figure 5. Visualizations of text-to-3D generation results. The two scores denote quality and alignment, respectively.

tively good, particularly for ProlificDreamer, Magic3D, and LatentNeRF. However, when additional surrounding information is incorporated or when multiple objects are placed, the quality metrics of all methods experience varying degrees of degradation.

In terms of alignment, some methods are able to reflect object information beyond the surroundings. This results in no significant decline in the *Single Object with Surroundings* set compared to the *Single Object* set. However, a noticeable decline is observed when the prompt set changes to *Multiple Objects*. This trend reflects the current issue with most works using Score Distillation Sampling (SDS) as guidance to supervise the generation of 3D scenes. Specifically, SDS is relatively stable for single objects, but when the descriptions of the surroundings are added or when there are multiple objects in the scene, the appearance of the surroundings may have many possibilities after denoising steps. There may be more possibilities for relative positions between multiple objects, leading to increased variability in the results generated by the diffusion model. This in turn reduces the stability when supervising the generation of 3D scenes, resulting in a significant decline in the results.

In contrast, ProlificDreamer uses Variational Score Distillation (VSD) instead of SDS. By optimizing the distribution of the scene rather than directly optimizing the rendering results of the scene for 3D generation, ProlificDreamer demonstrates a clear advantage in complex scenarios.

Parallels and contrasts of the quality and alignment metrics. It is worth noting that quality and alignment are not entirely correlated. Quality is more concerned with the geometry and subjective quality within a certain range, while alignment focuses on accurately restoring the information in the prompt. It is relatively sensitive to additional erroneous information, encouraging the generation of precise and clear 3D scenes.

For instance, the overall performance of Fantasia3D decreases markedly when generating multiple objects, as it fails to create precise 3D geometry, resulting in poor alignment compared to LatentNeRF. However, the quality of some generated objects is commendable with the obtained rich texture, making the overall quality higher than LatentNeRF.

ProlificDreamer typically generates more realistic textures, contributing to its superior quality. However, it sometimes generates a large amount of information not mentioned in the prompt, resulting in the possibility that the information described in the prompt only occupies a small part of the 3D generation results. Sometimes it only appears in the form of partial texture without significant geometry, which reduces its alignment index. Moreover, this characteristic is not what subsequent applications of text-to-3D want to see, further highlighting the importance of the alignment metric.

4.3. 2D Guidance Analysis

The majority of current text-to-3D methods utilize 2D priors associated with Stable Diffusion [24] for the generation of 3D scenes. To further investigate the efficacy of 2D guidance and the proficiency of current text-to-3D methods in harnessing this guidance for 3D generation, we examine the correlation between the 2D image generation quality of the diffusion model and the quality of the final 3D generation result. For each prompt in T^{3}Bench, we apply Stable Diffusion for text-to-image conversion and score the resulting images over the prompt using ImageReward. Notably, the text-to-3D methods employ view-dependent prompting during the generation with 2D guidance of the diffusion model. Descriptions of viewing angles (e.g. front view, side view) are added at the end of the prompt. Given that the range and granularity of viewing descriptions in view-dependent prompting vary across different text-to-3D methods, we directly use the original prompt without view-dependent prompting in the text-to-image generation. We then compute the correlation between this result and the quality metric (Multi-view capturing with ImageReward) result for the generated 3D scene.

	Single Obj.	Single Obj. with Surr.	Multi Obj.
Dreamfusion	0.211	0.184	0.045
Magic3D	0.229	0.158	0.059
LatentNeRF	0.290	0.191	0.050
Fantasia3D	0.159	0.153	0.006
SJC	0.228	0.159	0.040
ProlificDreamer	0.357	0.272	0.147

Table 3. The Spearman's ρ correlation between Stable Diffusion's 2D image generation quality and text-to-3D methods' generation qualities, averaged over all prompts.

Tab. 3 displays the Spearman correlation between the text-to-image scores for the 2D guidance and the final text-to-3D scores. It can be observed that that all correlations are relatively low, and there are two overall trends: 1) methods demonstrating better performance in text-to-3D also have higher correlation coefficients; and 2) when using different prompt sets, the correlation coefficient also follows the trend of *Single Object* greater than *Single Object with Surroundings*, and the latter greater than *Multiple Objects*. We attribute these outcomes to the fact that Stable Diffusion can generate satisfactory 2D images most of the time, even for complex prompts. However, 2D guidance may not be effectively used by text-to-3D methods — they may fail to generate accurate 3D scenes even though the 2D images are acceptable, leading to a low text-to-3D score while high text-to-image score. In addition, the 2D guidance may not be view-consistent, which does not significantly affect the text-to-image scores but can indeed lead to poorer quality in the final 3D generation. Superior methods like ProlificDreamer can better utilize 2D images to form a 3D scene, as suggested by its higher correlation, and as a result, can generate higher quality 3D scenes. These observations suggest that the current bottleneck of text-to-3D lies in the process of learning 3D from 2D guidance, and the view consistency of 2D guidance, rather than the generative capability of Stable Diffusion itself.

4.4. Multi-view Inconsistency Analysis

The Janus problem, or multi-view inconsistency issue, arises when using Stable Diffusion for guidance, as it may not always generate accurate front, side, or back views for training. Consequently, this can lead to errors in the generated 3D scenes, such as repeated 3D semantics from multiple angles, e.g., the pink piggy in Fig. 6. Employing regional convolution to evaluate the quality of a more global region, our multi-view quality metric is able to faithfully reflect the Janus problem within generated 3D scenes. Objects that manifest the Janus problem typically only score highly in a highly localized area, which would result in score decline after applying regional convolution.

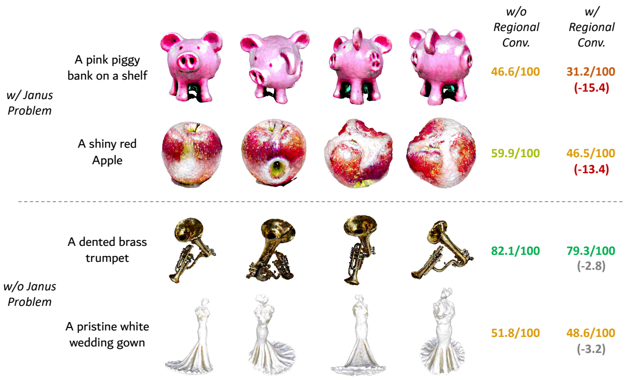

Figure 6. Variations in quality score alterations for objects, contingent upon the presence or absence of the Janus problem, following regional convolution.

To further investigate this, we conduct a case study. As illustrated in Fig. 6, we select examples both with and without the Janus problem. We then compare the scores of using regional convolution and not using it and directly taking the maximum value of all views. It is observed that objects with the Janus problem experience a significant decrease in the score derived with regional convolution, while such a decrease is not observed for objects without the Janus problem. This validates that the Janus problem can be reflected in our quality metric.

5. Conclusion

In this work, we present T^{3}Bench, the first comprehensive benchmark for evaluating text-to-3D generation methods. T^{3}Bench serves as a rich testbed as it provides diverse prompt suites, and supports fully automatic evaluation by incorporating our proposed multi-view quality and alignment metrics that closely correlate with human judgments. We carefully benchmark six prevalent text-to-3D methods on T^{3}Bench, and diagnose a number of common problems with current methods and problems specific to each of them.

6. Discussion

Size of Data. Unlike existing text-to-image methods that enable efficient generation, the current text-to-3D techniques are considerably slower, requiring a minimum of half an hour and potentially several hours for a single prompt. This makes it hard to test with larger sets of prompts.

Indirect Evaluation. Given the absence of an effective evaluation method that directly aligns the generated 3D scenes with human evaluation, there is an inevitable loss of information during the 3D to 2D rendering process, even with the efficacy of our multi-view capturing and processing scheme in evaluating geometry and other information. Likewise, there is no 3D captioning framework that matches the performance of BLIP in 2D image captioning. While the multi-view captioning and merging strategy we utilize typically generates accurate 3D captions, the merging process does not always yield flawless results.

References

[1] Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, et al. Benchmarking foundation models with language-model-as-an-examiner. *arXiv preprint arXiv:2306.04181*, 2023. 5

[2] Eslam Mohamed Bakr, Pengzhan Sun, Xiaoqian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. *arXiv preprint arXiv:2304.05390*, 2023. 3

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 3

[4] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023. 2, 5

[5] Jaemin Cho, Abhay Zala, and Mohit Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models, 2023. 3

[6] Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. threestudio: A unified framework for 3d content generation. https://github.com/threestudio-project/threestudio, 2023. 5

[7] Yuze He, Peng Wang, Yubin Hu, Wang Zhao, Ran Yi, Yong-Jin Liu, and Wenping Wang. Mmpi: a flexible radiance field representation by multiple multi-plane images blending, 2023. 1

[8] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 3

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2, 3

[10] Susung Hong, Donghoon Ahn, and Seungryong Kim. Debiasing scores and prompts of 2d diffusion for robust text-to-3d generation. *arXiv preprint arXiv:2303.15413*, 2023. 2, 3

[11] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 3, 5

[12] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 300–309, 2023. 1, 2, 5

[13] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 3, 5

[14] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 3

[15] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *arXiv preprint arXiv:2306.07279*, 2023. 5

[16] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures, 2022. 2, 5

[17] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 3

[18] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 conference papers*, pages 1–8, 2022. 1

[19] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3, 5

[20] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[21] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 5

[22] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023. 2

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3, 4

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 8

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3

[26] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 3

[27] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2, 3

[28] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 3

[29] Hoigi Seo, Hayeon Kim, Gwanghyun Kim, and Se Young Chun. Ditto-nerf: Diffusion-based iterative text to omnidirectional 3d model. *arXiv preprint arXiv:2304.02827*, 2023. 1

[30] Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937*, 2023. 1

[31] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021. 3

[32] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. *arXiv preprint arXiv:2304.12439*, 2023. 1

[33] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023. 5

[34] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 2, 3, 5

[35] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *arXiv preprint arXiv:2304.05977*, 2023. 2, 3, 4

[36] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. 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*, pages 20908–20918, 2023. 1

[37] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 1

[38] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019. 5

[39] Minda Zhao, Chaoyi Zhao, Xinyue Liang, Lincheng Li, Zeng Zhao, Zhipeng Hu, Changjie Fan, and Xin Yu. Efficientdreamer: High-fidelity and robust 3d creation via orthogonal-view diffusion prior. *arXiv preprint arXiv:2308.13223*, 2023. 2

7. Example Prompts

7.1. Question Generation

Single Object.

> *Please describe 20 objects' appearance for me in brief words, without background. Please make sure that the object you provided has enough diversity, and that the format is similar to my example. Here is an example: "A pig wearing a backpack".*

Single Object with Surroundings.

> *Please describe 20 objects for me in brief words. Please make sure that the object you provided has enough diversity, and that the format is similar to my example. Here is an example: "A black metal bicycle leaning against a brick wall".*

Multiple Objects.

> *Please describe 20 different scenes for me in brief words, each scene contains multiple objects. Do not describe the environment. Please make sure that the scenes you provided have enough diversity, and the format similar to my example. Here are examples: "A child with a red shirt is playing with a dog", or "Two coffee cups stand on the table".*

7.2. Multiple Caption Merging

> *Given a set of descriptions about the same 3D object, distill these descriptions into one concise caption. The descriptions are as follows:*
>
> *view1: ...*
> *view2: ...*
> ...
> *view$\{N\}$: ...*
>
> *Avoid describing background, surface, and posture. The caption should be:*

7.3. LLM Likert Scale Scoring

> *You are an assessment expert responsible for prompt-prediction pairs. Your task is to score the prediction according to the following requirements:*
>
> *1. Evaluate the recall, or how well the prediction covers the information in the prompt. if the prediction contains information that does not appear in the prompt, it should not be considered as bad.*
>
> *2. If the prediction contains correct information about color or features in the prompt, you should also consider raising your score.*
>
> *3. Assign a score between 1 and 5, with 5 being the highest. Do not provide a complete answer; give the score in the format: 3*
>
> *Prompt: ...*
>
> *Prediction: ...*

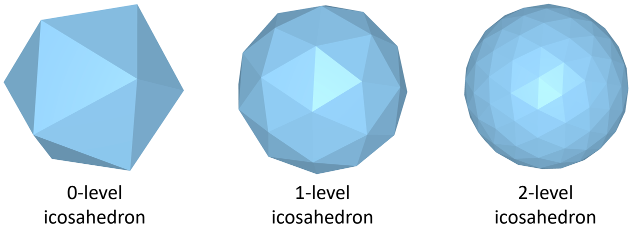

Figure 7. Schema of icosahedrons with different levels.

8. Experimental Details

8.1. Metric Evaluation

For the evaluation of metrics, we randomly select 30% of the prompts from each prompt set, along with their corresponding 3D mesh generated by the text-to-3D method. This results in a total of 540 samples. We request human annotators to carefully check the mesh in an interactive 3D viewer and score the responses on a scale of 1-5, based on their 3D quality and alignment. Below, we provide the annotation instructions:

> 1. Scoring is based on two dimensions: quality (which assesses the subjective quality of the 3D generation) and alignment (which evaluates how well the generated content covers the original prompt). These two dimensions are scored on a scale of 1 to 5, with 1 being the lowest and 5 the highest.
> 2. Please drag each generated mesh to our specified 3D viewer. After carefully examining the mesh from various angles, assign your score based on the above two dimensions.

8.2. Capture Viewpoint Selection

In order to uniformly select the capturing location of the 3D mesh, we construct the icosahedron and use its vertices as the location for the captures. The vertex coordinates of a level-0 unit icosahedron are computed as follows:

$$V^{(0)} = \sqrt{1+\phi^2} \cdot \begin{vmatrix} \phi & 1 & 0 \\ -\phi & 1 & 0 \\ \phi & -1 & 0 \\ -\phi & -1 & 0 \\ 1 & 0 & \phi \\ 1 & 0 & -\phi \\ -1 & 0 & \phi \\ -1 & 0 & -\phi \\ 0 & \phi & 1 \\ 0 & -\phi & 1 \\ 0 & \phi & -1 \\ 0 & -\phi & -1 \end{vmatrix}, \tag{2}$$

where

$$\phi = \frac{1+\sqrt{5}}{2}, \tag{3}$$

and there is an edge between every two points with a distance of $2/\sqrt{1+\phi^2}$, resulting in 12 vertices, 30 edges, and 20 triangle faces.

A level-K unit icosahedron can be obtained recursively by adding an extra vertice on every edge of a level-(K-1) unit icosahedron and adding an edge between every two new vertexes with a triangle face of the level-(K-1) unit icosahedron, then scaling every new vertex's coordinate to a length of 1. A demonstration of different level icosahedrons is shown in Fig. 7.

8.3. Capturing Poses Derivation

Since many text image scoring models are sensitive to rotation, we need to make sure that the angle of the shot is as free as possible from 2D rotation around the look-at vector. We ensure this constraint with the following procedure:

Given the location v of the shot, we can get the look-at vector as follows:

$$\mathbf{l} = -\frac{\mathbf{v}}{||\mathbf{v}||}. \tag{4}$$

Then, we acquire the horizontal vector $\mathbf{r}$ of the camera plane by

$$\mathbf{r} = \frac{\mathbf{u} \times \mathbf{l}}{||\mathbf{u} \times \mathbf{l}||}, \tag{5}$$

where $\mathbf{u}$ is the unit vector parallel with the positive direction of the vertical axis. The up vector of the camera plane can be calculated by

$$\mathbf{u}' = \mathbf{l} \times \mathbf{r}. \tag{6}$$

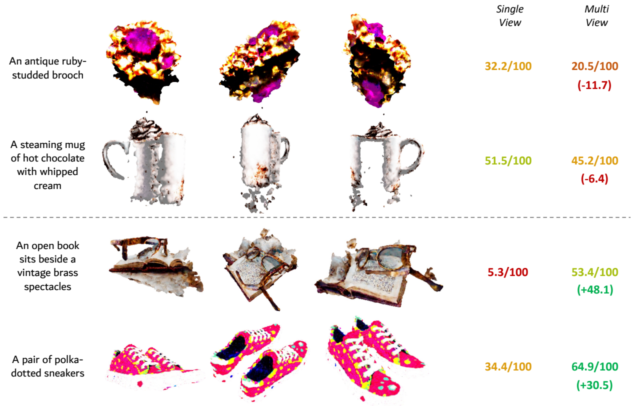
Single View
Multi View
An antique ruby-studded brooch
32.2/100
20.5/100
(-11.7)
A steaming mug of hot chocolate with whipped cream
51.5/100
45.2/100
(-6.4)
An open book sits beside a vintage brass spectacles
5.3/100
53.4/100
(+48.1)
A pair of polka-dotted sneakers
34.4/100
64.9/100
(+30.5)

Figure 8. Comparisons of the scoring between single-view capturing and our multi-view capturing scheme. The first image column denotes the single front view, and the other two image columns are captured from other viewpoints.

Finally, the camera matrix $\mathbf{P}$ is formed with

$$\mathbf{P} = [-\mathbf{r} \quad \mathbf{u}' \quad \mathbf{l} \quad \mathbf{v}]. \tag{7}$$

9. More Case Studies

9.1. Single-view vs. Multi-view Capturing

We further illustrate through a case study that adhering to the previous method and only capturing single-view images does not yield satisfactory evaluations. As depicted in Fig. 8, the first two examples demonstrate good subjective quality in the front view. However, their geometries are relatively poor, and there are noticeable residuals or artifacts when they are converted to other viewpoints. These can be identified with our multi-view capturing mechanism, which subsequently adjusts the scores accordingly. In the next two examples, the front view is partially obscured, which fails to fully represent the subjective quality of the generated objects. Our multi-view capturing mechanism can detect this and improve their scores accordingly.

9.2. More results

We provide more test prompts with generations and evaluations of different text-to-3D methods in Figs. 9, 10, 11.

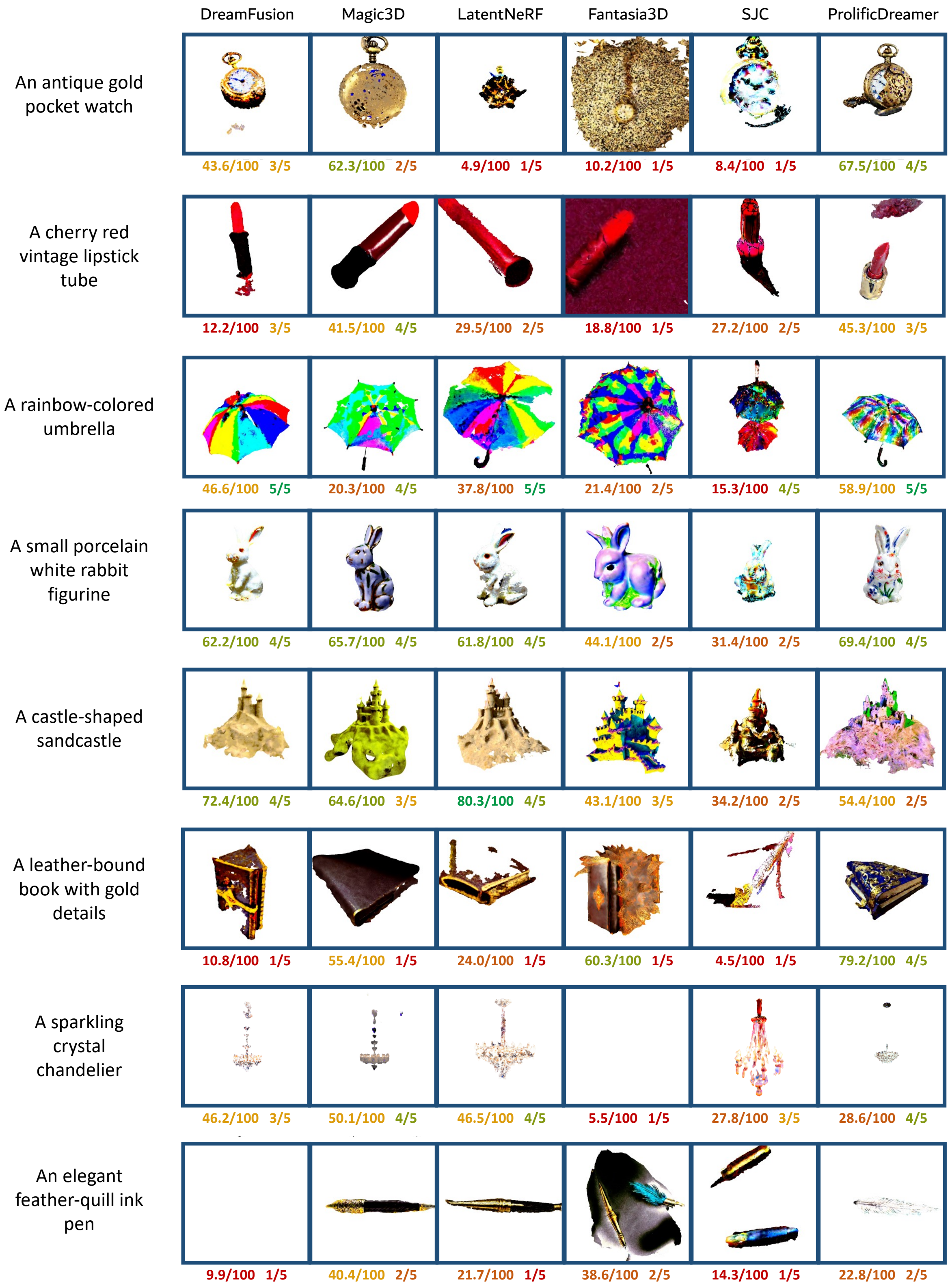

Figure 9. More results of our test prompts, including generations and evaluations of different text-to-3D methods (#1).

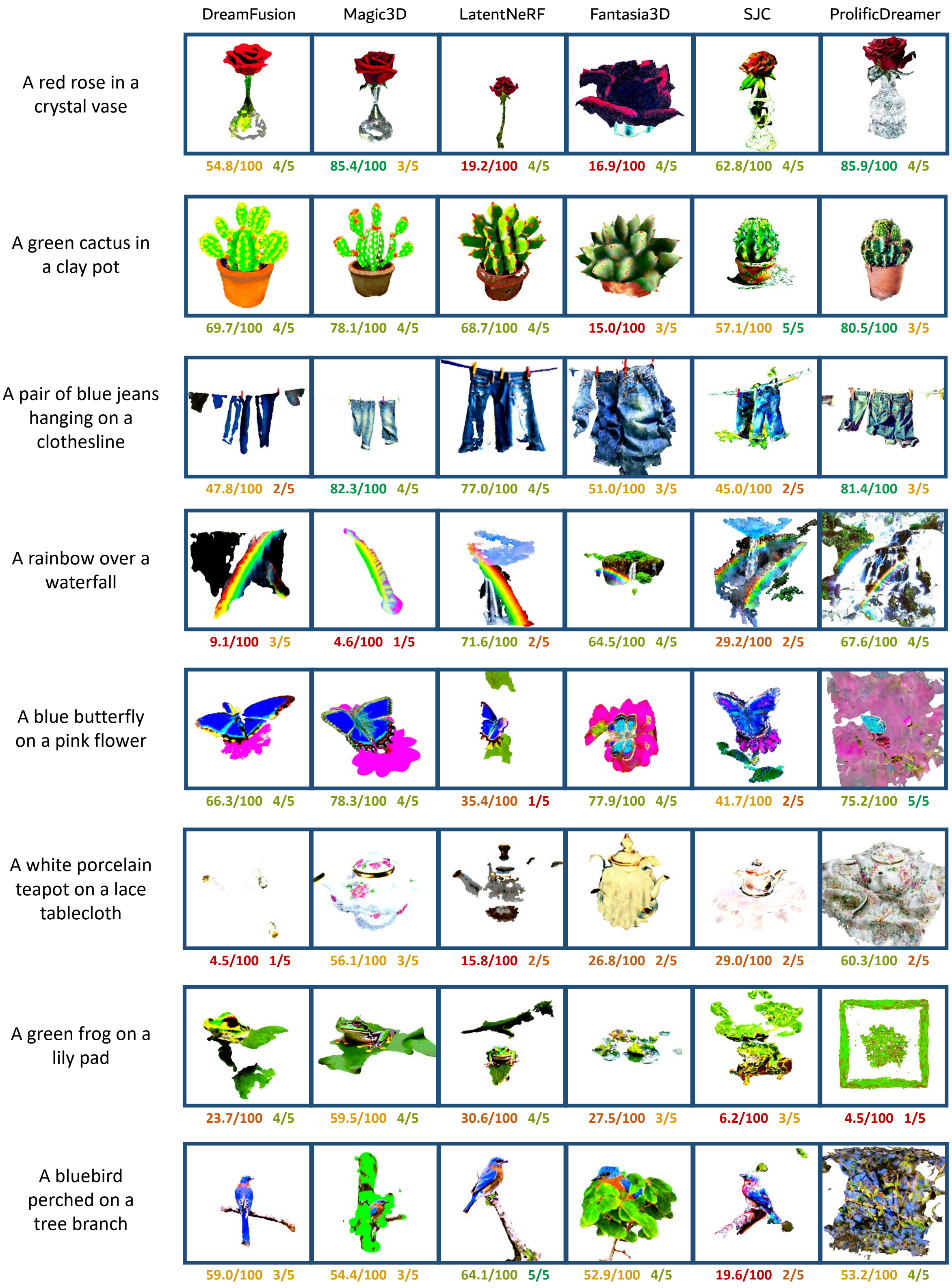

Figure 10. More results of our test prompts, including generations and evaluations of different text-to-3D methods (#2).

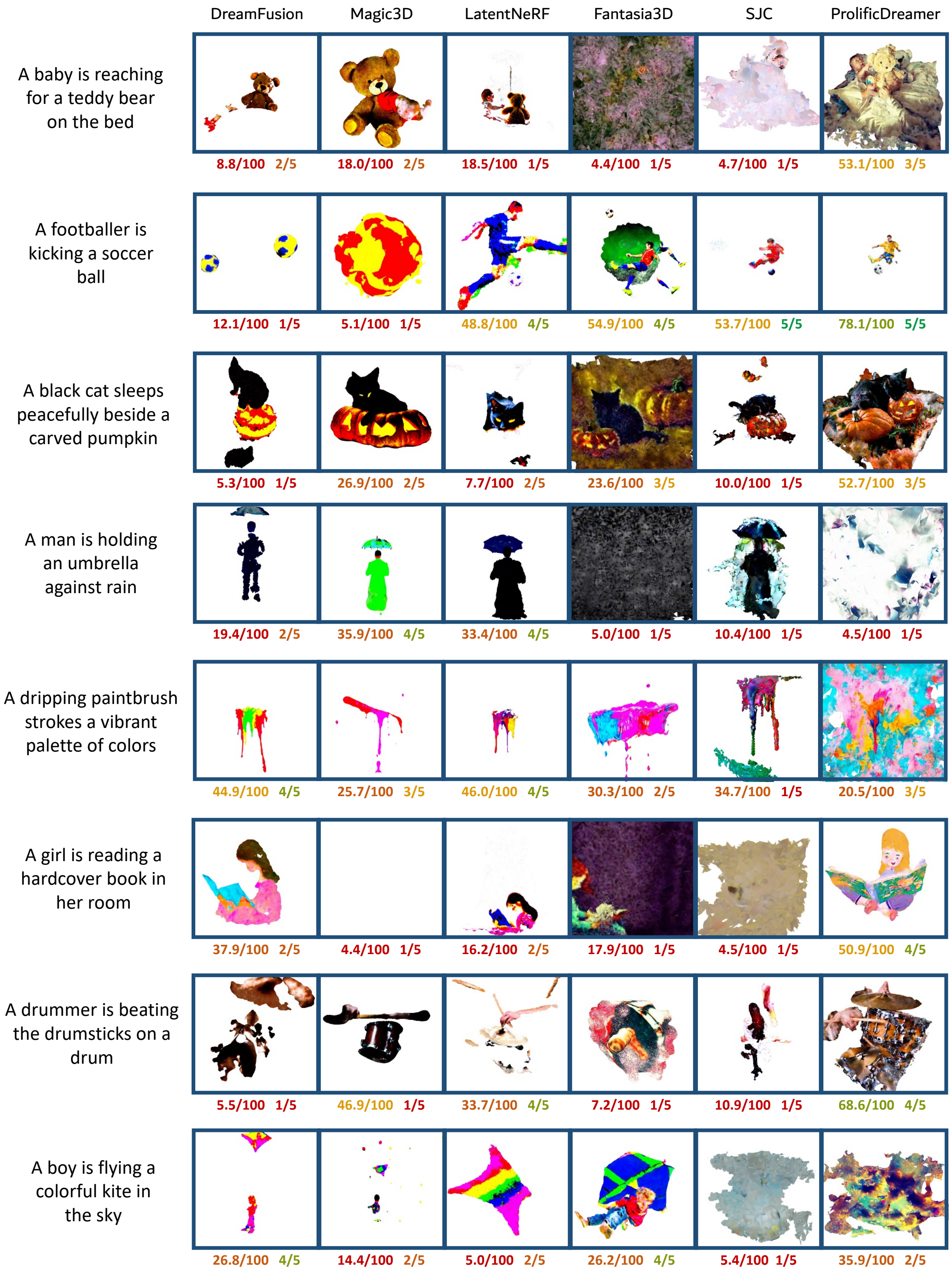

Figure 11. More results of our test prompts, including generations and evaluations of different text-to-3D methods (#3).