

GPT-4V(ision) is a Human-Aligned Evaluator for Text-to-3D Generation

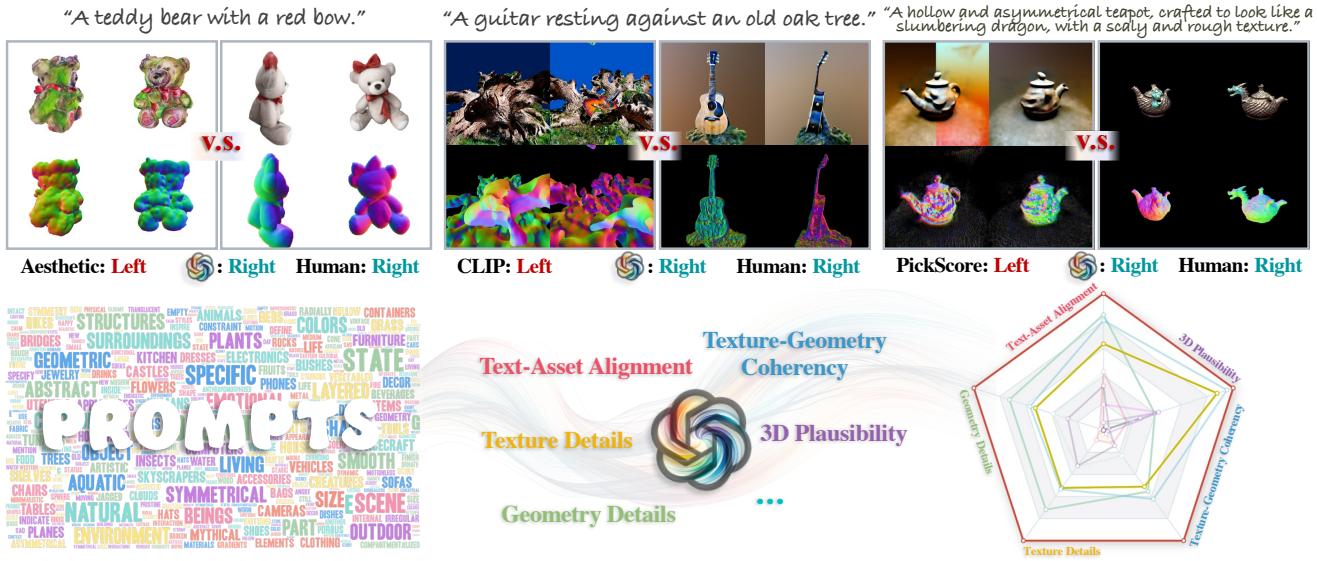
Tong Wu^{1,5*}Guandao Yang^{2*}Zhibing Li^{1,5*}Leonidas Guibas²Dahua Lin^{1,5}Kai Zhang³Ziwei Liu⁴¹ The Chinese University of Hong Kong² Stanford University³ Adobe Research⁴ S-Lab, Nanyang Technological University⁵ Shanghai Artificial Intelligence Laboratory

Figure 1. We present a versatile and human-aligned evaluation metric for text-to-3D generative methods. To this end, we design a prompt generator that can produce a set of input prompts targeting an evaluator’s demands. Moreover, we leverage GPT-4V to compare two 3D shapes according to different evaluation criteria. Our method provides a scalable and holistic way to evaluate text-to-3D models.

Abstract

Despite recent advances in text-to-3D generative methods, there is a notable absence of reliable evaluation metrics. Existing metrics usually focus on a single criterion each, such as how well the asset aligned with the input text. These metrics lack the flexibility to generalize to different evaluation criteria and might not align well with human preferences. Conducting user preference studies is an alternative that offers both adaptability and human-aligned results. User studies, however, can be very expensive to scale. This paper presents an automatic, versatile, and human-aligned evaluation metric for text-to-3D generative models. To this end, we first develop a prompt generator using GPT-4V to generate evaluating prompts, which serve as input to compare text-to-3D models. We further design a method instructing GPT-4V to compare two 3D assets according to user-defined criteria. Finally, we use these pairwise comparison results

to assign these models Elo ratings. Experimental results suggest our metric strongly aligns with human preference across different evaluation criteria. Our code is available at <https://github.com/3DTopia/GPTEval3D>.

1. Introduction

The field of text-to-3D generative methods has seen remarkable progress over the past year, driven by a series of breakthroughs. These include advancements in neural 3D representations [42, 47], the development of extensive datasets [10, 14, 15], the emergence of scalable generative models [24, 56, 62], and the innovative application of text-image foundational models for 3D generation [48, 51]. Given this momentum, it’s reasonable to anticipate rapidly increasing research efforts and advancements within the realm of text-to-3D generative models.

Despite recent advances, the development of adequate evaluation metrics for text-to-3D generative models has not

* Equal contribution.

kept pace. This deficiency can hinder progress in further improving these generative models. Existing metrics often focus on a single criterion, lacking the versatility for diverse 3D evaluation requirements. For instance, CLIP-based metrics [29, 51] are designed to measure how well a 3D asset aligns with its input text, but they may not be able to adequately assess geometric and texture detail. This lack of flexibility leads to misalignment with human judgment in evaluation criteria the metric is not designed for. Consequently, many researchers rely on user studies for accurate and comprehensive assessment. Although user studies are adaptable and can accurately mirror human judgment, they can be costly, difficult to scale, and time-consuming. As a result, most user studies have been conducted on a very limited set of text-prompt inputs. This leads to a question: *Can we create automatic metrics that are versatile for various evaluation criteria and align closely with human judgment?*

Designing metrics that meet these criteria involves three essential capabilities: generating input text prompts, understanding human intention, and reasoning about the three-dimensional physical world. Fortunately, Large Multimodal Models (LMMs), particularly GPT-4Vision (GPT-4V) [46], have demonstrated considerable promise in fulfilling these requirements [71]. Drawing inspiration from humans’ ability to perform 3D reasoning tasks using 2D visual information under language guidance, we posit that GPT-4V is capable of conducting similar 3D model evaluation tasks.

In this paper, we present a proof-of-concept demonstrating the use of GPT-4V to develop a customizable, scalable, and human-aligned evaluation metric for text-to-3D generative tasks. Building such an evaluation metric is similar to creating an examination, which requires two steps: formulating the questions and evaluating the answers. To effectively evaluate text-to-3D models, it is crucial to obtain a set of input prompts that accurately reflect the evaluators’ needs. Relying on a static, heuristically generated set of prompts is insufficient due to the diverse and evolving nature of evaluator demands. Instead, we developed a “meta-prompt” system, where GPT-4V generates a tailored set of input prompts based on evaluation focus. Following the generation of these input text prompts, our approach involves comparing 3D shapes against user-defined criteria, akin to grading in an exam. We accomplish this through designing an instruction template, which can guide GPT-4V to compare two 3D shapes per user-defined criterion. With these components, our system can automatically rank a set of text-to-3D models by assigning each of these models an Elo rating. Finally, we provide preliminary empirical evidence showing that our proposed framework can surpass existing metrics in achieving better alignment with human judgment in a diverse set of evaluation criteria. Results suggest that our metric can efficiently provide an efficient and holistic evaluation of text-to-3D generative models.

2. Related Work

Text-to-3D generation. Text-to-image generation models have become increasingly powerful with text-to-3D extensions being the next frontier (see [48] for a recent survey). However, due to limited amounts of 3D data, text-to-3D has mainly been driven by methods based on optimizing a NeRF representation [42]. For example, Dreamfusion [51] optimizes a NeRF using score-distillation-sampling-based (SDS) loss. The quality of such optimization-based methods [11, 37, 41, 51, 60, 63, 66, 68], however, is far behind that of text-to-image models [50, 54–56]. Compared with their 2D counterparts, they are generally lacking diversity, texture fidelity, shape plausibility, robustness, speed, and comprehension of complex prompts. On the other hand, Point-E [44] and Shap-E [30] train feed-forward 3D generative models on massive undisclosed 3D data. Though they show promising results with fast text-to-3D inference, their generated 3D assets look cartoonish without geometric and texture details. Recently, we notice a rapid change in the landscape of text-to-3D methods [38, 39] mainly due to the public release of the large-scale Objaverse datasets [16, 17]. Feed-forward methods trained on these datasets, e.g., Instant3D [36], have managed to make a big jump in text-to-3D quality, narrowing the performance gap between 3D and 2D generation. As we expect to see continuing progress in this area, it is critical to have robust evaluation metrics closely aligning with human judgment to measure different aspects of 3D generative models, including shape plausibility and texture sharpness. Such an evaluation metric can provide meaningful guidance for model design choices and support fair comparisons among the research community.

3D Evaluation Metrics. Evaluating 3D generative models is inherently challenging, requiring an understanding of both physical 3D worlds and user intentions. Traditional methods for evaluating unconditional or class-conditioned 3D models typically measure the distance between distributions of generated and reference shapes [1, 5, 9, 21, 40, 70]. However, these metrics are not readily applicable to text-conditioned generative tasks due to the difficulty in obtaining a comprehensive reference set, given the vastness of natural language inputs [6]. To alleviate this issue, prior work tried to curate a set of text prompts to evaluate key aspects of text-conditioned generative tasks [22, 51]. Our work complements this effort by creating a text-prompt generator using language instruction. Additionally, prior studies utilized multimodal embeddings, such as CLIP [52] and BLIP [34, 35], to aid the evaluation. For instance, the CLIP Similarity metric [29, 51] employs CLIP embeddings to assess text-to-3D alignment. However, these metrics are often tailored to measure specific criteria, lacking the flexibility to adapt to different requirements of text-to-3D evaluation. User preference studies are considered the gold stan-

dard for evaluating text-to-3D models, as adopted by many papers [5, 26, 37, 53, 58, 63]. While user studies offer versatility and accuracy, they are costly, time-consuming, and difficult to scale. Our automatic metrics can serve as an alternative to user preference studies, aligning well with human preferences while offering high customizability.

Large multimodality models. Following the success of large language models (LLMs) [3, 8, 12, 25, 46, 64], the focus has shifted to large multimodal models (LMMs) as the next frontier in artificial intelligence. Initial efforts of LMM involve combining computer vision with LLMs by fine-tuning visual encoders to align with language embeddings [2, 4, 18, 28, 34, 35, 65] or converting visual information to text [27, 59, 67, 72]. Most of these models are usually limited in scale. Recently, GPT-4V [45] has risen as the leading LMMs, benefiting from training on an unprecedented scale of data and computational resources. These LMMs have demonstrated a range of emerging properties [71], including their capability as evaluators for language and/or vision tasks [23, 74, 75]. In our work, we explore the use of GPT-4V in evaluating 3D generative models, a relatively under-explored application because GPT-4V cannot directly consume 3D information.

3. Method Overview

The goal of our evaluation metric is to rank a set of text-to-3D models based on user-defined criteria. Our method consists of two primary components. First, we need to decide which text prompt to use as input for the evaluation task. Toward this goal, we develop an automatic prompt generator capable of producing text prompts with customizable levels of complexity and creativity (Sec. 4). The second component is a versatile 3D assets comparator (Sec. 5). It compares a pair of 3D shapes generated from a given text prompt according to the input evaluation criteria. Together, these components allow us to use the Elo rating system to assign each of the models a score for ranking (Sec. 5.3).

4. Prompt Generation

Creating evaluation metrics for text-to-3D generative models requires deciding which set of input text prompts we should use as input to these models. Ideally, we would like to use all possible user input prompts, but this is computationally infeasible. Alternatively, we would like to build a generator capable of outputting prompts that can mimic the actual distribution of user inputs. To achieve this, we first outline the important components of an input prompt for text-to-3D models (Sec 4.1). Building on these components, we design a “meta-prompt” to instruct GPT-4V how to leverage these components to generate an input text prompt for text-to-3D models (Sec 4.2).

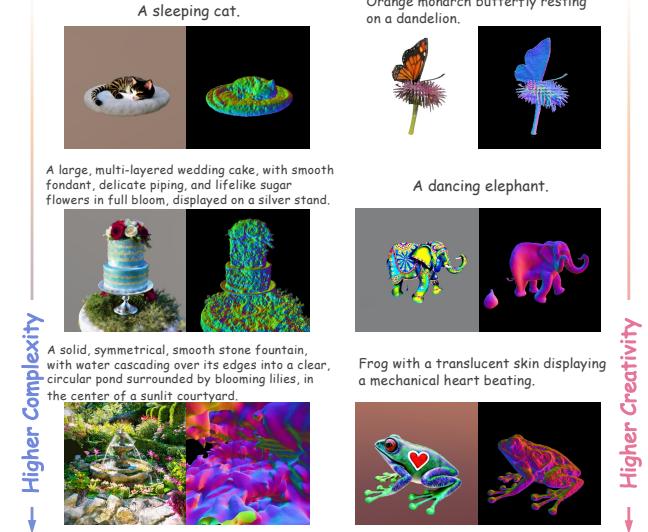


Figure 2. **Controllable prompt generator.** More complexity or more creative prompts often lead to a more challenging evaluation setting. Our prompt generator can produce prompts with various levels of creativity and complexity. This allows us to examine text-to-3D models’ performance in different cases more efficiently.

4.1. Prompt components

A typical input text prompt for text-to-3D models contains three components: subjects, properties, and compositions. Subjects usually involve nouns referring to objects or concepts the user would like to instantiate in 3D. “Cats”, “fire”, and “universe” are all examples of subjects. Properties include adjectives a user can use to describe the subjects or their interactions, such as “mysterious” and “weathered”. Finally, users will compose these concepts and properties together into a sentence or clause. The composition varies from as simple as joining different subjects and/or properties together with commas or as thoughtful as writing it as a grammatically correct and fluent sentence. In this work, we prompt GPT-4V to create a comprehensive list of words for subjects and properties. This list of subjects and properties will be used as building blocks to construct the “meta-prompt”, which is an instruction for GPT-4V to generate input text-prompts by composing these building blocks. Section B.1 contains more implementation details.

4.2. Meta-prompt

Provided with ingredients to create input prompts, we now need to automatically compose these ingredients together according to the evaluator-specified requirements. This requires the prompt generator to understand and follow the evaluator’s instruction. In this paper, we use GPT-4V’s ability to generate prompts following instructions. Specifically, we would like to build a text instruction asking GPT-4V to create a list of prompts that can be used as input for

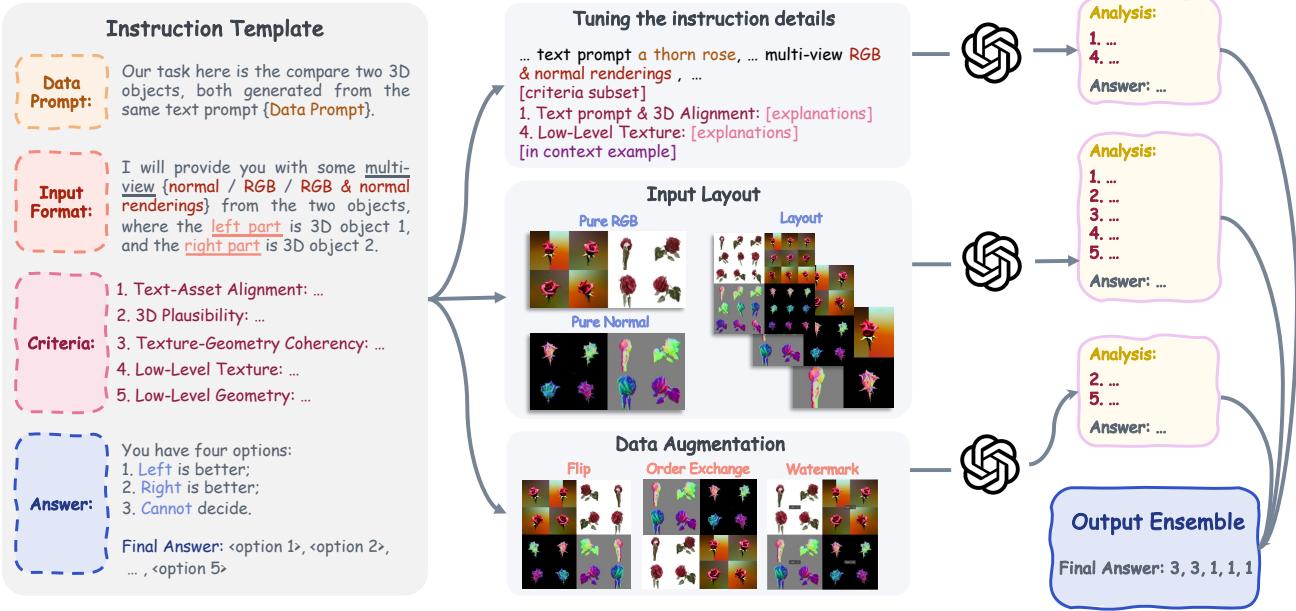


Figure 3. **Illustration of how our method compares two 3D assets.** We create a customizable instruction template that contains necessary information for GPT-4V to conduct comparison tasks for two 3D assets (Sec. 5.1). We complete this template with different evaluation criteria, input 3D images, and random seeds to create the final 3D-aware prompts for GPT-4V. GPT-4V will then consume these inputs to output its assessments. Finally, we assemble GPT-4V’s answers to create a robust final estimate of the task (Sec. 5.2)

text-to-3D models. We coin this instruction “meta-prompt”.

In order for GPT-4V to output prompts for text-to-3D models, we first provide GPT-4V with the necessary ingredients, *i.e.* a list of subjects and properties from the previous section. In addition to these, the meta-prompt needs to include a description of how the evaluator wants the output prompt set to be. For example, the evaluator might want to focus on complex prompts containing multiple subject interactions and properties, testing a text-to-3D models’ ability to generate complex objects. One might also be curious about these models’ performance in creative prompts involving subjects and descriptions that are not commonly seen in the real world. How complex and creative the input prompt can influence how challenging the evaluation task is. These two axes, complexity and creativity, are examples of evaluator’s criteria. Such criteria should be specified as language instructions attached to the “meta-prompt” along with all the ingredients. With both the prompt ingredient and the evaluator’s criteria properly included in the meta-prompt, our GPT-4V-based prompt generator can now compose sentences that adhere to the evaluator’s requirement. The appendix contains more details about our meta-prompt and prompt generation pipeline (Sec B.1).

Figure 2 shows prompts outputted from our generator with instruction asking for different complexity and creativity. We can see that high complexity introduces a larger number of objects, multifaceted descriptions, and occasion-

ally, a completely broken scene. Similarly, more creative prompts combine subjects, verbs, or adjectives in unconventional ways. Text-to-3D models also tend to struggle with these more creative prompts, failing to follow the description of these input prompts exactly. This suggests that input prompts distribution can greatly affect how challenging the evaluation task is. Being able to control the distributions of the input prompt allows us to examine the performance of these text-to-3D models through a more focused lens.

5. 3D Assets Evaluator

Now we can sample a set of text prompts, $\mathcal{T} = \{t_i\}_i$, using our generator. In this section, we will evaluate the performance of a set of text-to-3D generative models using \mathcal{T} as input prompts. Given a set of these models, $\mathcal{M} = \{M_j\}_j$, we use each model to generate one or more 3D shapes for each prompt. This results in a set of tuples: $\{(M_k, t_k, M_j(t_k, \mathbf{z}_k)) | M_k \in \mathcal{M}, t_k \in \mathcal{T}\}_k$, where \mathbf{z}_k represents the random noise influencing the shape generation. Our objective is to rank the text-to-3D models in \mathcal{M} based on a user-defined criterion. To accomplish this, we first prompt GPT-4V to compare two 3D assets generated from the same input text prompt (Sec 5.1 and Sec 5.2). We then use these pairwise comparison results to assign each of the models an Elo rating reflecting its performance (Sec 5.3).

5.1. Pairwise Comparison

At the core of our evaluation metric is the ability to answer the following question: *given a text prompt t , and two 3D shapes generated from two different models, say M_i and M_j , which 3D shape is better according to the evaluation criteria?* As discussed in previous sections, we hypothesize that one can leverage GPT-4V to achieve this task. However, since GPT-4V is trained on language and visual data, it lacks the ability to analyze 3D shapes directly. Therefore, our input to GPT-4V should include both text instructions and 2D visual renderings that can capture 3D information.

Specifically, for each of the two 3D assets, we will create a large image containing renderings of the 3D asset from four or nine viewpoints. These two images will be concatenated together before passing into GPT-4V along with the text instructions. GPT-4V will return a decision of which of the two 3D assets is better according to the instruction.

Text instruction. We need to communicate three pieces of information for GPT-4V to compare two 3D assets: instructions to complete a 3D comparison task, the evaluation criteria, and descriptions of the output format. We found it important to emphasize that the provided images are renders from different viewpoints of a 3D object. In addition to a plain description of the user-defined evaluation criteria, providing instruction about what kind of image features one should use when analyzing for a particular criteria is also useful. Finally, instead of requesting only the answer of which shape is better directly, we also prompt GPT-4V to explain how it arrives at its conclusion [7, 69].

Visual features of 3D shapes. Once GPT-4V has been prompted to understand the evaluation criteria and task of interest, we now need to feed the 3D shape into the GPT-4V model. Specifically, we need to create images that can convey the appearance and the geometry features of the 3D shapes. To achieve that, for each 3D object, we create image renders of the object from various viewpoints. For each of these viewpoints, we also render a surface normal image. These normal surface renders will be arranged in the same layout as the RGB render before being fed into GPT-4V. Using world-space surface normal renders leads to better results because they provide geometric information about the surface and allow reasoning for correspondence between views. Appendix B.2 has more implementation details.

5.2. Robust Ensemble

Even though GPT-4V is able to provide an answer to the pairwise shape comparison problem, its response to the same input can vary from time to time due to the probabilistic nature of its inference algorithm. In other words, we can view our GPT-4V 3D shape comparator’s outputs as a categorical distribution, and each response is a sample from the distribution. As a result, a single response from GPT-4V

might not capture its true prior knowledge since it can be affected by the variance during sampling. This is particularly the case when the variance of the output distribution is high (*e.g.*, when both choices are equally likely). Note that this is not a weakness of GPT-4V as similar situations can happen to human annotators when two objects are equally good according to a criterion. In other words, we are not interested in sampling one instance of how GPT-4V would make a decision. Instead, estimating with what probability GPT-4V will choose this answer is more useful.

One way to estimate such probability robustly from samples with variance is through ensembling, a technique that has also been explored in other tasks [71]. Specifically, we propose to ensemble outputs from multiple slightly perturbed inputs. The key is to perturb input prompts to GPT-4V without changing the task or evaluation criteria. The input includes the text instruction, visual images, as well as the random seed. Our methods deploy different perturbations, including changing random seeds, the layout of renders, the number of rendered views, and the number of evaluation criteria. Figure 3 illustrates how we perturb the input and ensemble the results from these perturbed inputs together. Appendix D includes more details.

5.3. Quantifying Performance

We have now obtained a list of comparisons among a set of models \mathcal{M} . The comparisons are over a variety of sampled prompts denoted as \mathcal{T} according to the user-defined criteria. Our goal is now to use this information to assign a number for each model in \mathcal{M} such that it best explains the observed result. Our quantification method should consider the fact that the comparison results are samples from a probability distribution, as discussed in the previous subsection.

This problem is commonly studied in rating chess players, where a game between two players can have different outcomes even if one player is better than the other. In chess and many other competitions, the Elo score [19] is perhaps the most widely adapted method to produce a numerical estimation that reflects players’ performance. The Elo rating system has also been adapted in prior works to evaluate image generative models [43, 61]. In this paper, we adapt the version proposed by Nichol et al. [43]. Specifically, let $\sigma_i \in \mathbb{R}$ denote the Elo score of the i^{th} model in \mathcal{M} . A higher score σ_i indicates better performance. We assume that the probability of model i beats model j is:

$$\Pr(\text{"}i \text{ beats } j\text{"}) = \left(1 + 10^{(\sigma_j - \sigma_i)/400}\right)^{-1}. \quad (1)$$

Our goal is to find score σ_i that can best explain the observed comparison results given the abovementioned assumption. This can be achieved via maximum likelihood estimation. Specifically, let A be a matrix where A_{ij} denotes the number of times model i beats model j in the list

Table 1. **Alignment with human judgment (higher is better).** Here we present Kendall’s tau ranking correlation [31] between rankings provided by a metrics and those provided by human experts. Higher correlation indicates better alignment with human judgment. We **bold-face** the most aligned method and underline the second place for each criterion. Our method achieves top-two performances for all evaluation criteria, while prior metrics usually only do well for at most two criteria.

Methods	Alignment	Plausibility	T-G Coherency	Tex Details	Geo Details	Average
PickScore [33]	0.667	<u>0.484</u>	0.458	0.510	0.588	0.562
CLIP-S [23]	0.718	0.282	0.487	0.641	0.667	0.568
CLIP-E [23]	<u>0.813</u>	0.426	0.581	0.529	0.658	0.628
Aesthetic-S [57]	0.795	0.410	<u>0.564</u>	0.769	<u>0.744</u>	<u>0.671</u>
Aesthetic-E [57]	0.684	0.297	0.555	0.813	0.684	0.611
Ours	0.821	0.641	<u>0.564</u>	0.821	0.795	0.710

Table 2. **Pairwise rating agreements (higher is better).** We measure the average probability that the decision of the metric matches that of human’s for each comparison. Our method achieves strong alignment across most criteria.

Metrics	Align.	Plaus.	T-G.	Text.	Geo.	Avg.
PickS.	0.735	<u>0.721</u>	0.713	0.690	<u>0.740</u>	0.720
CLIP	0.726	0.644	0.678	0.703	<u>0.715</u>	0.693
Aest.	<u>0.798</u>	0.698	0.753	<u>0.817</u>	0.780	<u>0.769</u>
Ours	0.810	0.826	<u>0.729</u>	0.843	0.735	0.789

of comparisons. The final Elo score for this set of models can be obtained by optimizing the following objective:

$$\sigma = \arg \min_{\sigma} \sum_{i \neq j} A_{ij} \log \left(1 + 10^{(\sigma_j - \sigma_i)/400} \right). \quad (2)$$

In this paper, we initialize $\sigma_i = 1000$ and then use the Adam optimizer [32] to minimize the loss to obtain the final Elo score. Please refer to Sec B.3 for more mathematical intuition about the formulation of the Elo score.

6. Results

In this section, we provide a preliminary evaluation of our metric’s alignment with human judgment across different criteria. We first introduce the experiment setup. We will discuss the main alignment results in Sec. 6.1. We then explore how to use our metric to evaluate different models holistically in Section 6.2. Finally, we briefly showcase how to extend our models to different criteria in Section 6.3.

Text-to-3D generative models to benchmark. We involve 13 generative models in the benchmark, including ten optimization-based methods and three recently proposed feed-forward methods. Please refer to Sec C for the complete list. We leverage each method’s official implementations when available. Alternatively, we turn to Threestudio’s implementation [20]. For methods designed mainly for image-to-3D, we utilize Stable Diffusion XL [49] to

generate images conditioned on text as input to these models. All experiments are conducted with default hyperparameters provided by the code.

Baselines metrics. We select three evaluation metrics with various considerations. **1) CLIP similarity** measures the cosine distance between the CLIP features [52] of the multi-view renderings and the text prompt. This metric is chosen because it is widely used in previous works as the metric for text-asset alignment [26, 29, 51]. **2) Aesthetic score** [57] is a linear estimator on top of CLIP that predicts the aesthetic quality of pictures. We choose this because it is trained on a large-scale dataset. **3) PickScore** [33] is a CLIP-based scoring function trained on the Pick-a-Pic dataset to predict human preferences over generated images. To compute the metrics above, we uniformly sample 30 RGB renderings for each of the generated assets. The CLIP similarity and aesthetic score can be directly computed from the multi-view renderings and averaged for each prompt. Since PickScore takes paired data as input for comparison, we assign 30 paired renderings for each pair of objects before averaging the PickScore results.

Evaluation criteria. While our method can potentially be applied to all user-defined criteria, in this work we focus on the following five criteria, which we believe are important for current text-to-3D evaluation tasks. **1) Text-asset alignment:** how well a 3D asset mirrors the input text description. **2) 3D plausibility:** whether the 3D asset is plausible in a real or virtual environment. A plausible 3D asset should not contain improbable parts such as multiple distorted faces (Janus problem) or noisy geometry floaters. **3) Texture details:** whether the textures and appearance of the shape are realistic, high resolution, and have appropriate saturation levels. **4) Geometry details:** whether the geometry makes sense and contains appropriate details. **5) Texture-geometry coherency:** whether geometry and textures agree with each other. For example, eyes of a character should be on reasonable parts of the face geometry.

Expert annotation. To evaluate the performance of our method, we need to conduct user preference studies to ob-

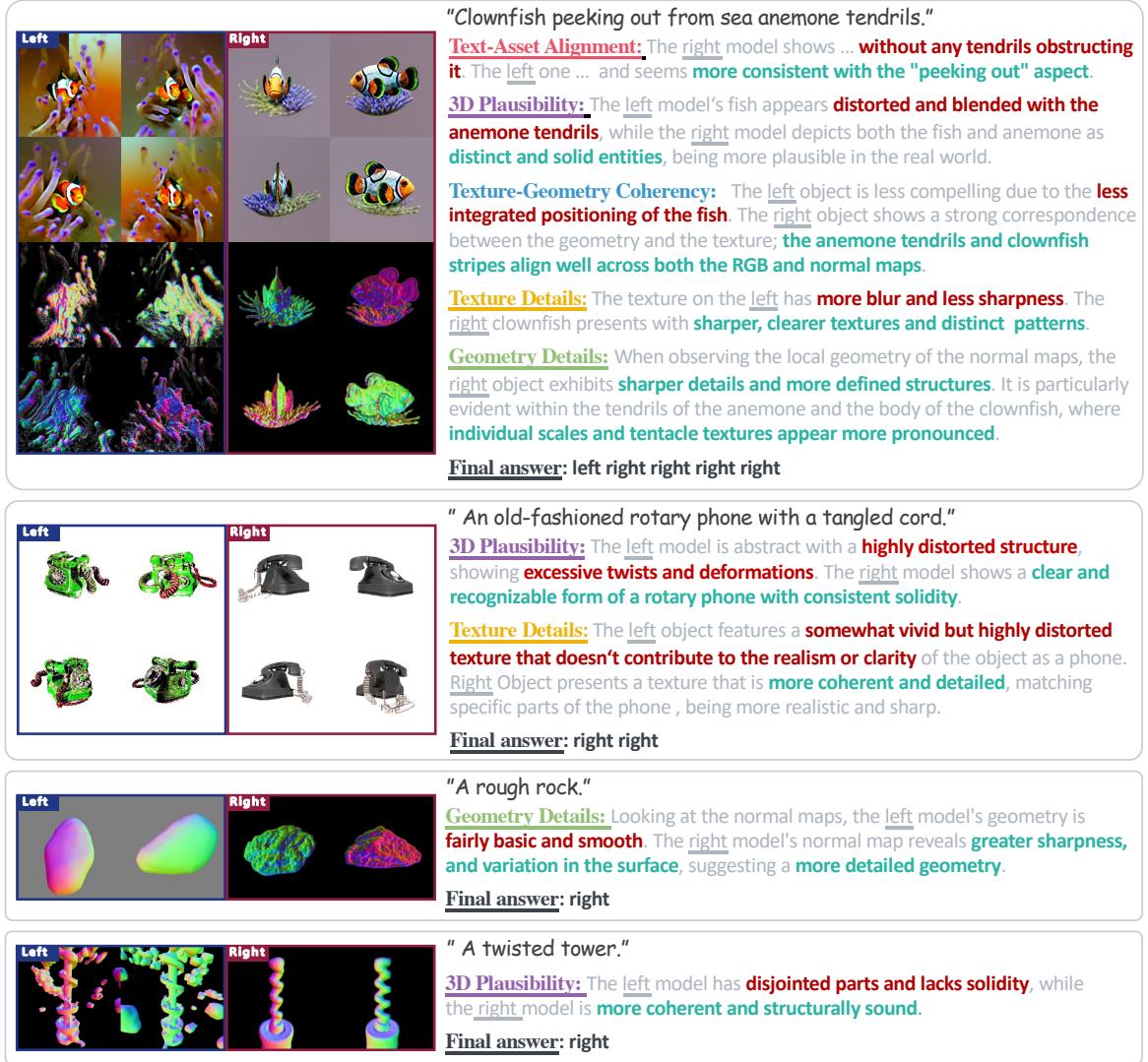


Figure 4. Examples of the analysis by GPT-4V. Given two 3D assets, we ask GPT-4V to compare them on various aspects and provide an explanation. We find that GPT-4V’s preference closely aligns with that of humans.

tain ground truth preference data. Our user studies will present the input text prompt alongside a pair of 3D assets generated by different methods for the same input. The user will be asked to identify which 3D asset satisfies the criteria of interest better. We recruited 20 human experts who are graduate students experienced in computer vision and graphics research to annotate the data. We assigned 3 annotators per comparison question per evaluation criteria. We compute a reference Elo rating using the formula in Sec. 5.3 using all expert annotations.

6.1. Alignment with Human Annotators.

In this section, we evaluate how well our proposed metric aligns with human preference. To achieve that, we use each metric to assign a score for each text-to-3D model for each evaluation criteria. Then, we compute Kendall’s tau

correlation [31] between the scores computed by the metrics and the reference scores. Table 1 shows the ranking correlations between scores predicted by different evaluation metrics and the reference Elo scores computed from expert annotators. We can see that our metrics achieve the best correlation in 4 out of 5 criteria, as well as the best average correlation. Note that our method achieves consistent performance across different criteria, while prior metrics usually perform well in only one or two. This highlights that our method is versatile in different evaluation criteria.

Our metric also shows strong human correlation for each 3D asset comparison question, which is a harder task. To measure that, we assume the response to each comparison question follows a Bernoulli distribution with probability p to select the first shape. Let p_i be the probability that the evaluation metric will select the first shape at question

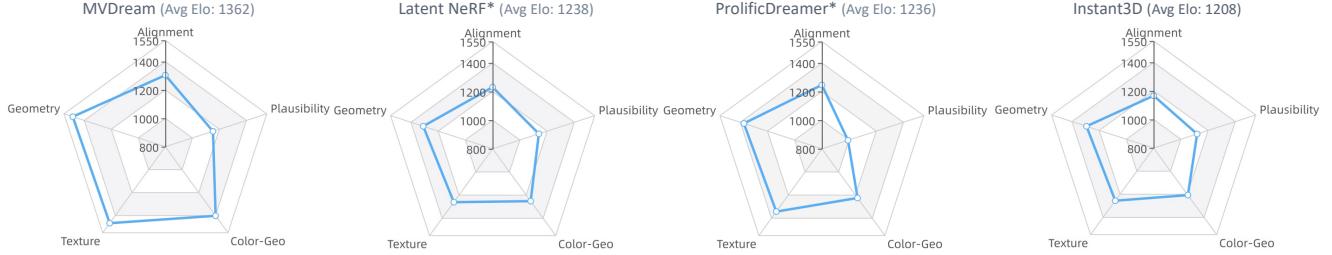


Figure 5. Holistic evaluation. Since our evaluation metric is human-aligned in multiple criteria, we can evaluate text-to-3D models more holistically. In this figure, we listed the Radar charts of the top four text-to-3D models according to their averaged Elo scores across all five criteria we evaluated. The Radar charts report the Elo rating for each of the five criteria. These radar charts can provide relative strengths and weaknesses among these models, providing guidance to improve these models. * indicates results from Threestudio implementation.

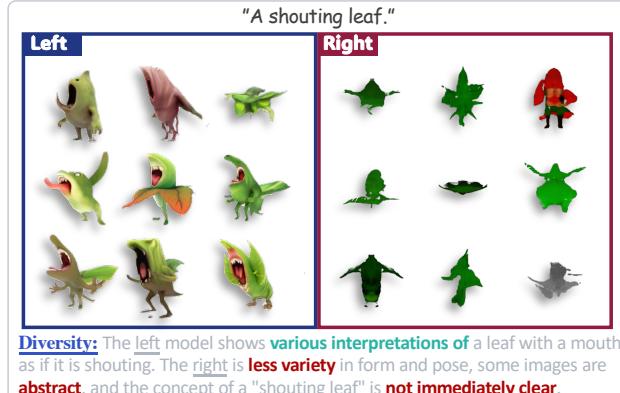


Figure 6. Diversity evaluation. Our method can be extended to evaluate which text-to-3D models output more diverse 3D assets.

i and q_i be that of a human annotation. We measure the pairwise rating agreement using the probability of a random sample from the metric agreeing with that from a human: $\frac{1}{N} \sum_{i=1}^N p_i q_i + (1 - p_i)(1 - q_i)$. Table 2 shows that our method achieves top-two agreement across all but one criteria. Figure 4 shows some exemplary outputs from our method. We can see that GPT-4V is also able to provide some analysis justifying its final choice.

6.2. Holistic Evaluation

The versatility of our method lands the provision to paint a holistic picture of each text-to-3D model’s performance. Specifically, we compute each model’s average Elo scores across each criterion and present the Radar charts of the models achieving the top averaged Elo scores in Figure 5. According to our metric, MVDream [60] won first place on all five criteria. MVDream achieves about 100 more ELO scores than its runner-ups. The second, the third, and the fourth places are taken by Latent-NeRF [41], ProlificDreamer [68], and Instant3D [36]. These models achieve similar averaged Elo scores, with differences of less than 30 Elos. These three models achieve about 100 Elos more

than the next tiers of models, which score about 1100 Elos.

While Latent-NeRF, ProlificDreamer, and Instant3D achieve similar overall scores, our metrics allow further analysis into the relative strengths and weakness of each models. For example, ProlificDreamers show strong performance in three criteria: alignment, geometry details, and texture details. However, its performance in 3D Plausibility is lagging behind when comparing with the other top-performing models. Among these three models, Instant3D [36] is a feed-forward method that takes a much shorter time compared to the top two methods. While our evaluation metrics’ reliability can still be limited, we hope that such a holistic picture can provide essential guidance for developing future text-to-3D algorithms.

6.3. Extension to Other Criteria

While we focus our empirical studies in five criteria, our metric can be adapted to evaluating a different criteria users might care about. For example, it is important that a generative model can produce different outputs given different random seeds. This aspect is commonly overlooked by most text-to-3D metrics. With small modification of the text and image prompt input into GPT-4V, our method can be applied to evaluate diversity. Figure 6 shows the visual image we provide GPT-4V when prompting it to answer the question about which model’s output has more diversity. For each method, we produce 9 3D assets using different random seeds. We render each of these assets from a fixed camera angle to create the input image fed into GPT-4V. The text in Figure 6 is an excerpt of GPT-4V’s answer. We can see that GPT-4V is able to provide a reasonable judgment about which image contains renders of more diverse 3D assets. Currently, we are restricted to qualitative studies because most existing text-to-3D models are still compute-intensive. We believe that large-scale quantitative study is soon possible with more compute-efficient text-to-3D models, such as Instant3D, becoming available.

7. Discussion

In this paper, we have presented a novel framework leveraging GPT-4V to establish a customizable, scalable, and human-aligned evaluation metric for text-to-3D generative tasks. First, we propose a prompt generator that can generate input prompts according to the evaluator’s needs. Second, we prompt GPT-4V with an ensemble of customizable “3D-aware prompts.” With these instructions, GPT-4V is able to compare two 3D assets according to an evaluator’s need while remaining aligned to human judgment across various criteria. With these two components, we are able to rank text-to-3D models using the Elo system. Experimental results confirm that our approach can outperform existing metrics in various criteria.

Limitations and future work. While promising, our work still faces several unresolved challenges. First, due to limited resources, our experiment and user studies are done on a relatively small scale. It’s important to scale up this study to better verify the hypothesis. Second, GPT-4V’s responses are not always true. For example, GPT-4V sometimes shows hallucinations—a prevalent issue for many large pretrained models [73]. GPT-4V can also process some systematic errors, such as bias toward certain image positions [74, 75]. Such biases, if unknown, could induce errors in our evaluation metric. While our ensembling technique can mitigate these issues, how to solve them efficiently and fundamentally remains an interesting direction. Third, a good metric should be “un-gamable”. However one could potentially construct adversarial patterns to attack GPT-4V. This way one might gain a high score without needing to produce high-quality 3D assets. Last, while our method is more scalable than conducting user preference studies, we can be limited by computation, such as GPT-4V API access limits. Our method also requires a quadratically growing number of comparisons, which might not scale well when evaluating a large number of models when compute is limited. It would be interesting to leverage GPT-4V to intelligently select input prompts to improve efficiency.

Acknowledgement. This project was in part supported by Google, Samsung, Stanford HAI, Vannevar Bush Faculty Fellowship, ARL grant W911NF-21-2-0104, and Shanghai AI Lab. We would love to thank members of Stanford Computational Imaging Lab, Stanford Geometric Computation Group, Shanghai AI Lab, and Adobe Research for useful feedback and discussion.

References

- [1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Learning representations and generative models for 3d point clouds. In *International Conference on Machine Learning*, 2017. [2](#)
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sébastien Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022. [3](#)
- [3] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Tachard Passos, Siyamak Shakeri, Emanuel Taropa, Paige Bailey, Z. Chen, Eric Chu, J. Clark, Laurent El Shafey, Yanping Huang, Kathleen S. Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Michael Brooks, Michele Catasta, Yongzhou Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, C Crépy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, M. C. D’iaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fan Feng, Vlad Fienber, Markus Freitag, Xavier García, Sebastian Gehrmann, Lucas González, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, An Ren Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wen Hao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Mu-Li Li, Wei Li, Yaguang Li, Jun Yu Li, Hyeontaek Lim, Han Lin, Zhong-Zhong Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellar, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alexandra Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Marie Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniela Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Ke Xu, Yunhan Xu, Lin Wu Xue, Pengcheng Yin, Jiahui Yu, Qiaoling Zhang, Steven Zheng, Ce Zheng, Wei Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. Palm 2 technical report. *ArXiv*, abs/2305.10403, 2023. [3](#)
- [4] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Yitzhak Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models. *ArXiv*, abs/2308.01390, 2023. [3](#)
- [5] Sherwin Bahmani, Jeong Joon Park, Despoina Paschalidou, Xinguang Yan, Gordon Wetzstein, Leonidas J. Guibas, and Andrea Tagliasacchi. Cc3d: Layout-conditioned generation

- of compositional 3d scenes. *ArXiv*, abs/2303.12074, 2023. 2, 3
- [6] Eslam Mohamed Bakr, Pengzhan Sun, Xiaogian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20041–20053, 2023. 2
- [7] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gaminazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczek, et al. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*, 2023. 5
- [8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. 3
- [9] Eric Chan, Connor Z. Lin, Matthew Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, S. Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16102–16112, 2021. 2
- [10] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1
- [11] 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, 16
- [12] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2022. 3
- [13] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960. 17
- [14] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022. 1
- [15] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian LaForte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023. 1
- [16] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian LaForte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023. 2
- [17] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 2
- [18] Danny Driess, F. Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Ho Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Peter R. Florence. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, 2023. 3
- [19] Arpad E Elo. The proposed uscf rating system, its development, theory, and applications. *Chess Life*, 22(8):242–247, 1967. 5, 16
- [20] 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. three-studio: A unified framework for 3d content generation, 2023. 6, 17
- [21] Zekun Hao, Arun Mallya, Serge J. Belongie, and Ming-Yu Liu. GanCraft: Unsupervised 3d neural rendering of minecraft worlds. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14052–14062, 2021. 2
- [22] Yuze He, Yushi Bai, Matthieu Lin, Wang Zhao, Yubin Hu, Jenny Sheng, Ran Yi, Juanzi Li, and Yongjin Liu. T3bench: Benchmarking current progress in text-to-3d generation. *ArXiv*, abs/2310.02977, 2023. 2
- [23] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation met-

- ric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 3, 6
- [24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1
- [25] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. Training compute-optimal large language models. *ArXiv*, abs/2203.15556, 2022. 3
- [26] Lukas Höllerin, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. *ArXiv*, abs/2303.11989, 2023. 3, 6
- [27] Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A. Smith, and Jiebo Luo. Promptcap: Prompt-guided task-aware image captioning. *ArXiv*, abs/2211.09699, 2022. 3
- [28] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. Language is not all you need: Aligning perception with language models. *ArXiv*, abs/2302.14045, 2023. 3
- [29] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022. 2, 6
- [30] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2, 16
- [31] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–93, 1938. 6, 7
- [32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CorR*, abs/1412.6980, 2014. 6
- [33] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *arXiv preprint arXiv:2305.01569*, 2023. 6
- [34] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 2022. 2, 3
- [35] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv*, abs/2301.12597, 2023. 2, 3
- [36] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. <https://arxiv.org/abs/2311.06214>, 2023. 2, 8, 16, 19
- [37] 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 (CVPR)*, pages 300–309, 2023. 2, 3, 16
- [38] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Learning to generate multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 2, 16
- [39] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3d: Single image to 3d using cross-domain diffusion, 2023. 2, 16
- [40] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. *arXiv: Machine Learning*, 2016. 2
- [41] Gal Metzger, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022. 2, 8, 16, 19
- [42] 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, 2
- [43] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 5, 16
- [44] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2, 16
- [45] OpenAI. Gpt-4v(ision) system card. *OpenAI*, 2023. 3, 18
- [46] R OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023. 2, 3
- [47] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 1
- [48] Ryan Po, Wang Yifan, and Vladislav Golyanik et al. State of the art on diffusion models for visual computing. In *arxiv*, 2023. 1, 2
- [49] Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *ArXiv*, abs/2307.01952, 2023. 6, 17
- [50] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2
- [51] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 1, 2, 6, 13, 16, 19

- [52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 2, 6
- [53] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan T. Barron, Yuanzhen Li, and Varun Jampani. Dreambooth3d: Subject-driven text-to-3d generation. *ArXiv*, abs/2303.13508, 2023. 3
- [54] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2
- [55] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.
- [56] 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, 2
- [57] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. *ArXiv*, abs/2210.08402, 2022. 6
- [58] Ho Youn Seo, Hayeon Kim, Gwanghyun Kim, and Se Young Chun. Ditto-nerf: Diffusion-based iterative text to omnidirectional 3d model. *ArXiv*, abs/2304.02827, 2023. 3
- [59] Zhenwei Shao, Zhou Yu, Mei Wang, and Jun Yu. Prompting large language models with answer heuristics for knowledge-based visual question answering. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 14974–14983, 2023. 3
- [60] Yichun Shi, Peng Wang, Jiangle Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023. 2, 8, 16, 19
- [61] Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. Improving image captioning with better use of captions. *arXiv preprint arXiv:2006.11807*, 2020. 5
- [62] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1
- [63] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *ArXiv*, abs/2309.16653, 2023. 2, 3, 16
- [64] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023. 3
- [65] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, Felix Hill, and Zacharias Janssen. Multimodal few-shot learning with frozen language models. In *Neural Information Processing Systems*, 2021. 3
- [66] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. *arXiv preprint arXiv:2212.00774*, 2022. 2, 16
- [67] Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, Shih-Fu Chang, Mohit Bansal, and Heng Ji. Language models with image descriptors are strong few-shot video-language learners. *ArXiv*, abs/2205.10747, 2022. 3
- [68] 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, 8, 16, 19
- [69] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022. 5, 13
- [70] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge J. Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 4540–4549, 2019. 2
- [71] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9, 2023. 2, 3, 5, 14, 16, 18
- [72] Andy Zeng, Adrian S. Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael S. Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Peter R. Florence. Socratic models: Composing zero-shot multimodal reasoning with language. *ArXiv*, abs/2204.00598, 2022. 3
- [73] Muru Zhang, Ofir Press, Will Merrill, Alisa Liu, and Noah A. Smith. How language model hallucinations can snowball. *ArXiv*, abs/2305.13534, 2023. 9
- [74] Xinlu Zhang, Yujie Lu, Weizhi Wang, An Yan, Jun Yan, Lianke Qin, Heng Wang, Xifeng Yan, William Yang Wang, and Linda Ruth Petzold. Gpt-4v (ision) as a generalist evaluator for vision-language tasks. *arXiv preprint arXiv:2311.01361*, 2023. 3, 9
- [75] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023. 3, 9

A. Overview

This supplementary material includes additional details about our experiment and methods, which can not be fully covered in the main paper due to limited space. We first provide more details about the method, including our meta-prompts and comparison prompts, in Section B. Experiment details, such as baselines and data, are included in Section C. We also provide detailed ablation studies about the effectiveness of different ways to prompt GPT-4V and ensemble its output (Section D). More experimental results are provided in Section E. Finally, we demonstrate some failure cases of our methods in Section F.

B. Method Details

We will include detailed descriptions about how we implement the two components of our method: prompt generator (in Sec. B.1) and 3D assets evaluator (in Sec B.2). Section B.3 provides additional details about how we use the elo rating.

B.1. Prompt Generator

Our prompt generation pipeline includes a conversation with GPT-4V. We aim to design a conversation pipeline that can provide GPT-4V with the necessary context of the prompt generation task while remaining customizable so that evaluators can adapt this conversation to their need.

Our first prompt describes the task of generating text prompt for text-to-3D generative models. For example, what’s the typical length of each prompt and how do we want the distribution of the final collection of prompts look like. We also include a description of how an evaluator might want to control this generator in this prompt. Please see Figure S7-S8 for this opening prompt.

Now that GPT-4V has some basic understanding of the task, we will further provide it with necessary components needed to composed the prompt. First, we provide with GPT-4V a list of ten main categories from which it’s able to select subjects of the prompt from (see Figure S1, first box). These categories are chosen to include the most common classes of objects users are interested in generating in text-to-3D tasks. The goal is to make our generator aware of a comprehensive list of potential options, reducing the risk of our prompt generator being unintentionally biased toward certain categories when performing in-context learning with exemplary prompts. These ten categories focuses on what one might wish to generate (*i.e.* the *subjects* mentioned in Section 4.1 of the main paper).

In addition to choosing what to generate, the text-to-3D generative model user also might want to specify a certain state the subject is in (*e.g.* “a sleeping cat” as opposed to “a cat”). Such description is referred to as the **properties** in Section 4.1 of the main paper. To achieve this, we addi-

tionally provide GPT-4V a list of descriptions of different properties a user of text-to-3d model might be interested in including (see Figure S1, second row, where each column include list of properties under one of the five aspects we listed). Note that this specific instruction will base on the subjects and a given level of creativity and complexity the evaluator specifies, which will be attached to the beginning of prompt. Please see Figure S7-S8 for the complete detailed prompt.

Finally, we will provide our prompt generator with a list of exemplary prompt where it can model the output from. Users can curate this list of prompt according to their need. In our case, we include the prompts by Dreamfusion [51].

With these exemplary prompts and our provided instruction designed to be customizable according to different scenarios, GPT-4V is now able to provide a list of prompts according to the evaluator’s input. Examples of the generated prompts from different levels of creativity and complexity are shown in Figure S2. We can see that users can create prompts with different difficulties and focus using our prompt generator. While we only focus on only two different axes (*i.e.* creativity and complexity) in this work, but our prompt generating pipeline can be easily adapted to other needs as the evaluators can change various part of our conversations accordingly.

B.2. 3D Assets Evaluator

In the instruction for the GPT-4V evaluator, we first explain the task and the definition of each criterion. Then, we asked to take a close look at the input pairs of multi-view images and normal maps before providing its analysis and final comparisons. Finally, we specify a specific output format for GPT-4V to follow. In our output format, we also require GPT-4V to provide a short reasoning of why it commit to certain choices, inspired by prompting technique such as chain-of-thought [69]. Specifying an output format can also help us better collect results at a larger scale. An example of such an instruction is shown in Figure S3.

Note that the description of the criteria can be adjusted according to evaluation task. In additional, one can also include in-context learning by breaking down this single text-prompt into multiple pieces, where the following pieces provide exemplary answer of the comparison task in hand. While we believe this can potentially provide a better performance, we only use a single text-prompt and a single image for each pairwise comparison due to limited API access.

In additional to the text instruction, we also need to render images from different view-points to input into GPT-4V. For each text-to-3D model, we provide one, four, or nine renders with camera evenly distributed surroundingly. Note that different text-to-3D generative models might use different coordinate systems, and GPT-4V might prefer certain

Subject						
Living Beings Household Items	Plants Clothing and Accessories	Buildings and Structures Abstract Objects	Vehicles	Furniture Natural Elements	Electronics Food and Beverages	
Appearance <ul style="list-style-type: none"> Color: Specific color, patterns, gradients, ... Materials: Wood, metal, glass, fabric, stone, ... Textures: Smooth, rough, furry, scaly, ... Finish: Glossy, matte, translucent, opaque, ... Size: Small, medium, large, specific dimensions, ... State: New, old, worn, pristine, ... 	Geometry <ul style="list-style-type: none"> Volume: hollow, solid, porous, or layered, ... Symmetry: symmetrical, asymmetrical, or radially symmetrical, ... Contours: smooth, jagged, irregular, or undulating, ... Internal Structures: empty, compartmentalized, or multi-layered, ... Shape: cone, cylinder, sphere, ... 	Status <ul style="list-style-type: none"> Static: Still, motionless, ... Dynamic: Moving, changing, ... Emotional State: Happy, sad, angry, ... Physical State: Broken, intact, in use, ... Interaction: Interacting with another object or environment, ... 	Scene <ul style="list-style-type: none"> Environment: Indoor, outdoor, urban, rural, natural, fantastical, ... Context: Part of a larger scene, event, or story, ... Lighting: Day, night, artificial, natural, shadows, highlights, ... Weather: Sunny, rainy, cloudy, stormy, ... Scale: The relative size of the object in the scene, ... 	Style <ul style="list-style-type: none"> Aesthetic: Minimalistic, ornate, modern, vintage, ... Cultural: Asian, African, Western, Middle Eastern, ... Emotional: Cheerful, gloomy, energetic, calm, ... Functional: Practical, decorative, symbolic, ... Conceptual: Abstract, realistic, surrealistic, impressionistic, ... 		

Figure S1. Subjects and properties provided for the prompt generator.

Creativity		Complexity				
High	Beach pebble with an intricate labyrinth pattern naturally engraved. Caterpillar with a keyboard pattern on its back. An origami crane made from a map. Zebra with barcode stripes.	An umbrella that blossoms with holographic constellations underneath when it rains. An ice cream scoop that serves up scoops of cloud fluff instead of ice cream. A velvet-lined violin case, which opens to reveal a garden of miniature roses.	A hammock strung between two skyscrapers, swaying high above a neon cityscape. An array of small, solid, symmetrical, pastel-colored eggs, each revealing a miniature, enchanted forest scene when cracked open.			
Medium	A quill pen, feather shifts through rainbow hues. A teddy bear, fur matted, one eye missing. An orange tabby cat shaped cookie jar. Floating bonsai tree, roots in mid-air.	A dragon-shaped kite, with scales that shimmer in the sunlight as it dances in the wind. An ensemble of jellyfish-like hanging lamps, pulsing with soft bioluminescence. A plush octopus whose arms are gently waving pencils.	Various hollow, asymmetrical, textured seashells, collected in a sand-filled, clear glass jar with a twine-tied neck, displayed on a windowsill. A large, hollow, asymmetrically shaped amphitheater, with jagged stone seating, nestled in a natural landscape, a classical play being performed as the sun sets.			
Low	Canvas sneakers, white, slightly dirt-stained. A weathered hiking backpack with patches. Denim jacket with patches and buttons. Knitted scarf, maroon and gold stripes. Ceramic mug with a chip on the rim.	A cluster of tents pitched near a forest, campfire smoke curling into the evening sky. A trio of spicy tacos with lime wedges on a plate, ready to be served. A worn leather recliner with a knitted throw draped over the back. Orange monarch butterfly resting on a dandelion. Gray squirrel with an acorn in its mouth.	A solid, symmetrical, smooth stone fountain, with water cascading over its edges into a clear, circular pond surrounded by blooming lilies, in the centre of a sunlit courtyard. An oversized, porous, sphere-shaped birdcage, made of woven golden wires, with a matte finish, housing a small, mechanical, singing bird that flutters in a lifelike manner. Several solid, spherical, weathered cannonballs, with a rough cast-iron texture, lying beside a rusted cannon in a historical fort overlooking a serene bay.			
Low		Medium		High		Complexity

Figure S2. Examples of the generated prompts with different levels of creativity and complexity.

views of the object than others. We leave it to the evaluator or the text-to-3D model developer to adjust the coordinate system to allow GPT-4V to make the best judgment.

Finally, we create several slightly perturbed version of

the input instruction and images to obtain multiple outcomes from GPT-4V. We will ensemble these outcomes to provide a more accurate estimate, as suggested by Yang et al. [71]. Specifically, we perturb the visual information

Our task here is to compare two 3D objects, both generated from the same text description. We want to decide which one is better according to the provided criteria.

Instruction

1. Text prompt and Asset Alignment. Focus on how well they correspond to the given text description. An ideal model should accurately reflect all objects and surroundings mentioned in the text prompt, capturing the corresponding attributes as described. Please first describe each of the two models, and then evaluate how well it covers all the attributes in the original text prompt.

2. 3D Plausibility. Look at both the RGB and normal images and imagine a 3D model from the multi-view images. Determine which model appears more natural, solid, and plausible. Pay attention to any irregularities, such as abnormal body proportions, duplicated parts, or the presence of noisy or meaningless 3D structures. An ideal model should possess accurate proportions, shapes, and structures that closely resemble the real-world object or scene.

3. Geometry-Texture Alignment. This examines how well the texture adheres to the geometry. The texture and shape should align with each other locally. For instance, a flower should resemble a flower in both the RGB and normal map, rather than solely in the RGB. The RGB image and its corresponding normal image should exhibit matching structures.

4. Low-Level Texture Details. Focus on local parts of the RGB images. Assess which model effectively captures fine details without appearing blurry and which one aligns with the desired aesthetic of the 3D model. Note that overly abstract and stylized textures are not desired unless specifically mentioned in the text prompt.

5. Low-Level Geometry Details. Focus on the local parts of the normal maps. The geometry should accurately represent the intended shape. Note that meaningless noise is not considered as high-frequency details. Determine which one has a more well-organized and efficient structure, which one exhibits intricate details, and which one is more visually pleasing and smooth.

6. Considering all the degrees above, which one is better overall?

Take a really close look at each of the multi-view images for these two 3D objects before providing your answer.

When evaluating these aspects, focus on one of them at a time.

Try to make independent decisions between these criteria.

Output format

To provide an answer, please provide a short analysis for each of the abovementioned evaluation criteria. The analysis should be very concise and accurate.

For each of the criteria, you need to make a decision using these three options:

1. Left (object 1) is better;
2. Right (object 2) is better;
3. Cannot decide.

IMPORTANT: PLEASE USE THE THIRD OPTION SPARSLY.

Then, in the last row, summarize your final decision by "<option for criterion 1> <option for criterion 2> <option for criterion 3> <option for criterion 4> <option for criterion 5> <option for criterion 6>".

Example

"

Analysis:

1. Text prompt and Asset Alignment: The left one ...; The right one ...; The left/right one is better or cannot decide.

2. 3D Plausibility. The left one ...; The right one ...; The left/right one is better or cannot decide.

3. Geometry-Texture Alignment. The left one ...; The right one ...; The left/right one is better or cannot decide.

4. Low-Level Texture Details. The left one ...; The right one ...; The left/right one is better or cannot decide.

5. Low-Level Geometry Details. The left one ...; The right one ...; The left/right one is better or cannot decide.

6. Overall, ... The left/right one is better or cannot decide.

Final answer:

x x x x x (e.g., 1 2 2 3 2 1 / 3 3 3 2 1 3 / 3 2 2 1 1 1)

"

Figure S3. An example of prompts used to guide the GPT-4V evaluator.

with three versions: including pure RGB renders, pure surface normal maps, and containing both. For text instructions, we experiment with three versions: 1) GPT-4V is required to evaluate a full list of six criteria, 2) include only the criteria of interest, and 3) explicitly ask GPT-4V to evaluate geometry first. We also perturb the number of render views in three versions: 1) only a single view-point; 2) four view-points in a 2x2 layout, and 3) 9 view-points in a 3x3 layout. Finally, we also augment the visual information in three ways: 1) horizontally flipping the left-right order of the objects; 2) vertically flipping the up-down order of the RGB images and normal maps and 3) adding visual watermark into the image [71]. A detailed ablation will be provided in the later section (Sec R1).

B.3. ELO Rating

Comparing two 3D assets can yield different outputs depending on the evaluator or the criteria. This is because human preference is highly subjective and stochastic. We need to model such stochasticity when designing a rating that correctly reflects how much a text-to-3D method is preferred by human evaluators. Such stochasticity is also encountered when two chess players are playing against each other. Since a stronger player can still lose to a weaker player when the stronger player he or she is having a bad day. In other words, the outcome of comparisons can be noisy. Fortunately, the ELO rating system is designed to create a rating that can reflect the true player’s performance while the comparison results are noisy. We will provide some basic intuition of how our ELO system worked for completeness and please refer to prior literatures for more details [19, 43].

The ELO rating system tries to assign a number, σ_i for the i^{th} player, which reflects the strength of this player. Let’s examine the case when two players, i and j , play together and assume without loss of generality that $\sigma_i > \sigma_j$. The probability of i^{th} player winning the game should be larger when the difference between σ_i and σ_j is larger. One way to define such probability of winning is the following:

$$\Pr(\text{"}i \text{ beats player } j\text{"}) = \frac{1}{1 + 10^{\frac{\sigma_j - \sigma_i}{c}}}, \quad (3)$$

where c controls the scale of the difference (*i.e.* if the difference is c then the probability of winning will be 1/11). From this formula, if $\sigma_i = \sigma_j$, then the probability that player i wins over player j is exactly 1/2. It’s also easy to see that

$$\Pr(\text{"}i \text{ beats } j\text{"}) + \Pr(\text{"}j \text{ beats } i\text{"}) \quad (4)$$

$$= \frac{1}{1 + 10^{\frac{\sigma_j - \sigma_i}{c}}} + \frac{1}{1 + 10^{\frac{\sigma_i - \sigma_j}{c}}} = 1. \quad (5)$$

During experiment, we observe that i beats j for A_{ij} number of times and j beats i for A_{ji} number of times. The

idea is to find the assignment of σ_i and σ_j such that it best reflects the empirical observation. Specifically, we achieve this by maximum likelihood estimation:

$$\begin{aligned} & \max_{\sigma_{i,j}} \Pr(A_{ij}, A_{ji} | \sigma_i, \sigma_j) \\ &= \max_{\sigma_{i,j}} \log \left(\Pr(\text{"}i \text{ beats } j\text{"})^{A_{ij}} \Pr(\text{"}j \text{ beats } i\text{"})^{A_{ji}} \right) \\ &= \max_{\sigma_{i,j}} A_{ij} \log (\Pr(\text{"}i \text{ beats } j\text{"})) + A_{ji} \log (\Pr(\text{"}j \text{ beats } i\text{"})) \\ &= \min_{\sigma_{i,j}} A_{ij} \log \left(1 + 10^{\frac{\sigma_j - \sigma_i}{c}} \right) + A_{ji} \log \left(1 + 10^{\frac{\sigma_i - \sigma_j}{c}} \right). \end{aligned}$$

Equation (1) in the main paper can be directly derived from the equation above by summing over all pairs of $i \neq j$. In practice, we will initialize $\sigma_i = 1000$ for all i and use Adam optimizer to optimize σ_i for this loss for 10000 iterations with a learning rate of 0.1. Since Elo score is invariant to adding or subtracting a constant, we further calibrate our scores by setting Dreamfusion [51] to have an Elo of 1000 for all criteria.

Note that this method consider the setting where the outcomes include only i wins or j wins. In our case, there are non trivial number of 3D assets pairs from which the human annotator cannot determine which is better. To handle these cases, we follow Nichol et al. [43] to add a win to both methods. This can effectively dilate the number of winning times. Our A_{ij} counts the number of times text-to-3D generative model i wins over model j over any captions. This can be adapted in this theoretical framework by considering the distribution of input text-prompts:

$$\Pr(\text{"}i \text{ beats player } j\text{"}) = \int \Pr(\text{"}i \text{ beats player } j\text{"}|t) P(t) dt,$$

where t denotes an input text prompt. Following most Elo systems, we choose $c = 400$.

C. Experimental Details

In this section we will provide some additional experiment details. We provide a detailed list of text-to-3d generative models for benchmarking in Section C.1. Additional detail about our user studies is provided in Section C.2.

C.1. Text-to-image Models

We involve 13 generative models in the benchmark, including ten optimization-based methods and three recently proposed feed-forward methods. Please refer to the supplementary for the complete list of methods. For optimized-base methods, we include DreamFusion [51], SJC [66], Latent-Nerf [41], Magic3D [37], Fantasia3D [11], Prolific Dreamer [68], DreamGaussian [63], MVDream [60], SyncDreamer [38] and Wonder3D [39]. For feed-forward methods, we include PointE [44], Shap-E [30], and Instant3D [36]. We leverage each method’s official imple-

mentations when available. Alternatively, we turn to Three-studio’s implementation [20]. For methods designed mainly for image-to-3D, we utilize Stable Diffusion XL [49] to generate images conditioned on text as input to these models. Experiments are conducted with default hyperparameters.

C.2. User study details

In this paper, we mainly rely on labels provided by expert annotators. Our expert annotators are all graduate students with computer graphic background (*e.g.* they all have experience looking at surface normal maps). We recruit twenty such expert annotators from this background. For 13 methods, we create pairwise comparison between each pair of methods (so 78 method pairs). For each method pairs of methods, we sample 3 captions from the caption generators. For each of these 234 comparisons, we assign three different experts to rank all criteria. The experts are asked to pick a shape that performs better according to certain criteria or indicate that these two shapes are indistinguishable. Each user will fill out a query form that includes the same description we provided with GPT-4V. Different from what we provided with GPT-4V, expert annotators are able to see video of 360 rotated render of the object in RGB and in surface normals. The annotator can control the video (*e.g.* replaying to certain point) to examine the 3D shape in more details. The video is rendered in 1024x2048 resolution. In our setting, expert annotators are provided with more information comparing to what we provided to GPT-4V. Our expert annotators achieve reasonable agreement with greater than 0.53 Cohen kappa [13].

One disadvantage of expert annotation is that it’s difficult to get by so we are only able to obtain a small scale of such annotation. On the contrary, one can obtain larger-scale annotation from general users. We’ve also conducted some preliminary exploration conducting user preference studies with general users. Unfortunately, we found that data collected from general users are very noisy. Specifically, we recruited about 53 users and performed the same user studies as done with the expert. In addition to the instruction provided with GPT-4V, we also provide examples comparison and requires the user to go through a short training session. The average user agreement (measured by Cohen’s kappa) among these users can barely reach 0.3. This can potentially be caused by the fact that general users do not have experience reasoning about 3D asset information, so computer graphics terms (*e.g.* texture, surface normal, or geometry), become harder to understand. As a result, we leverage expert user data as reference labels in this paper. How to conduct large-scale user studies with more available annotators remains a very interesting future work direction.

D. Ablation Studies

In this section, we will examine key technical design choices of our methods. Specifically, we will focus our ablation on different ways to perturb the input data to create ensemble results (Section 5.2 of main paper). We first carry out an ablation study on different ways to perturb the input (Section D.1). Then we show an ablation on how to ensemble these perturbations together (Section D.2).

Due to limited API access, we are not able to obtain enough GPT-4V queries to compute method-level human alignment scores as in Table 1 or the pair-level alignment score in Table 2 in the main papers. To address this limitation, we use an alternative metric in the ablation. For each variant of our methods, we randomly sample 78 different comparisons of 3D assets. For each comparison, we compute the probability that our GPT-4V 3D asset evaluator would select one shape versus the other. We denote such empirically estimated probability as p_i for our method’s variant to choose one shape at the i^{th} comparison. Let q_i be the same probability estimated from expert annotation data. We return the following L1-distance as an estimation of how our method is *mis*-alignment with human judgment:

$$\text{L1-dist}(p, q) = \frac{2}{N} \sum_{i=1}^N |p_i - q_i|, \quad (6)$$

where N is the number of comparisons here. Note that the lower this metric is, the better alignment with human judgement.

D.1. Ablation for GPT-4V Input Perturbation

We conduct ablations on different ways to perturb the inputs for GPT-4V. We will investigate four different categories of perturbations, including *visual information*, *text instruction*, *view number*, and *augmentation*. The main results are summarized in Table R1.

Visual Information The visual information can include RGB images, normal maps, or both of them. Purely using RGB as input can benefit the perception of texture details, probably because GPT-4V can spend more computational power on the presentation of the textures. However, not all the criteria can be evaluated merely from the RGB information. As a result, skip evaluation of alignment on those criteria, namely “Color-Geo” and “Geometry”. Only presenting the normal maps to GPT-4V does not bring much improvement for its alignment to human choices even for “Geometry Details”. We can see that RGB renders seem to play the most important role for GPT-4V to make human-aligned decisions. Surface normal render is required to perform many evaluations of criteria about geometries.

Table R1. **Ablation studies on different visual and textual input to GPT-4V.** We mark rank one, rank two, and rank three in each criterion with increasingly lighter shades of blue, and the same baseline (RGB + Normal, 2x2, Joint) is marked in gray.

Methods		Alignment (↓)	Plausibility (↓)	Color-Geo (↓)	Texture (↓)	Geometry (↓)
Visual Information	Pure RGB	0.523	0.564	-	0.354	-
	Pure Normal	0.674	0.654	-	-	0.579
	RGB + Normal	0.518	0.672	0.628	0.444	0.510
Text Instruction	Joint	0.518	0.672	0.628	0.444	0.510
	Separate	0.451	0.597	0.610	0.433	0.528
	Geo-first	0.682	0.646	0.662	0.487	0.505
View number	1	0.592	0.644	0.603	0.423	0.438
	2x2	0.518	0.672	0.628	0.444	0.510
	3x3	0.546	0.582	0.654	0.503	0.559
Augmentation	Horizontal Flip	0.615	0.702	0.676	0.522	0.651
	Vertical Flip	0.764	0.695	0.754	0.738	0.708
	Watermark	0.605	0.559	0.597	0.577	0.492

Text Instruction We experiment different way to input user criteria into the text instruction. Jointly inputting all criteria into a same text prompt can significantly reduce the number of API calls required. An example of this kind of text instruction can be seen in Figure S3. We also try to evaluate only one criterion at a time. One can see a clear improvement for most of the degrees thanks to the more focused analysis, especially for “Text-Asset Alignment”. This presents a trade-off between compute and accuracy.

View number The number of views denotes how many multi-view images are shown at the same time. Given the assumption that the visual context size is fixed for GPT-4V [45], this ablation explores the trade-off between the perception of global coherency and local details. Presenting only one view at a time can largely improve GPT-4V’s ability in evaluating low-level criteria like “Texture-Geometry Alignment”, “Texture Details”, and “Geometry Details”. However, the scarcity of views leads to challenges in evaluating the global criteria like “Text-Asset Alignment” and “3D Plausibility”. Increasing view numbers to four or nine will largely alleviate this problem.

Augmentation In the study, we have also experimented with various visual augmentation techniques [71], which refers to changing the visual input slightly without changing the key information contained in the image. We experiment with three augmentation methods: horizontal flipping of object positions (*i.e.* “Horizontal Flip”), the rearrangement of RGB images in conjunction with their respective normal maps (*i.e.* “Verticle Flip”), and the inclusion of watermark annotations to indicate the “left” and “right” objects (*i.e.* “Watermark”). Adding watermarks slightly improves the alignment. This can be a result of watermarks reduc-

ing the ambiguity happened when we refer to certain image positions from the text instruction.

Other findings. Another interesting finding is that the results get worse when the normal maps are emphasized in the inputs. For example, in the setting of “Geo-first”, we first provide the surface normal and ask questions involving geometries before providing the full RGB renders and asking the rest of questions. The setting of “Pure Normal” also emphasizes surface normal map by not including the RGB renders. These settings both lead to slightly worse result than the baseline method.

D.2. Ablation for Output Ensemble.

In this section, we want to explore what is the best way to combine different perturbations together. The exploration space of such combinations is huge as there are n^k numbers of ways to combine n ways to perturb the input together to create k ensembles. Interestingly, we observed that the variance of the L1-distance reduces as we increase the number of ensembles increases. This is depicted from Figure S4-(a) shows the distribution of the L1-distance when ensembling different number of results together. We can see that the alignment performance is not sensitive to particular choice of input perturbations when the number of ensembles is large enough. To achieve the most effective results, we will incorporate a combination of various view numbers, augmentations, visual information, and individual queries for specific criteria, selecting 4-5 strategies overall for the final combination. Figure S4-(b) that ensembling these strategies together results in a metric outperforms all the previous metrics when measured in L1-distance.

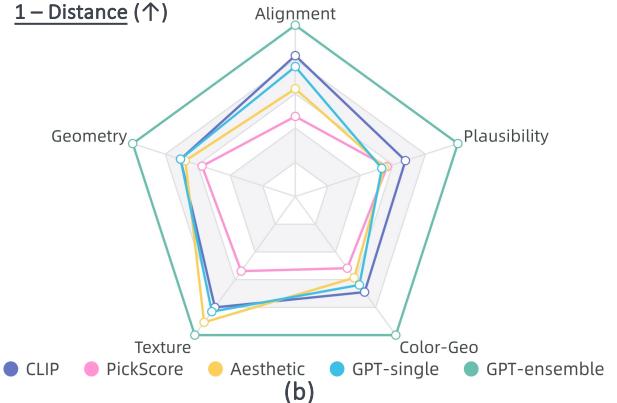
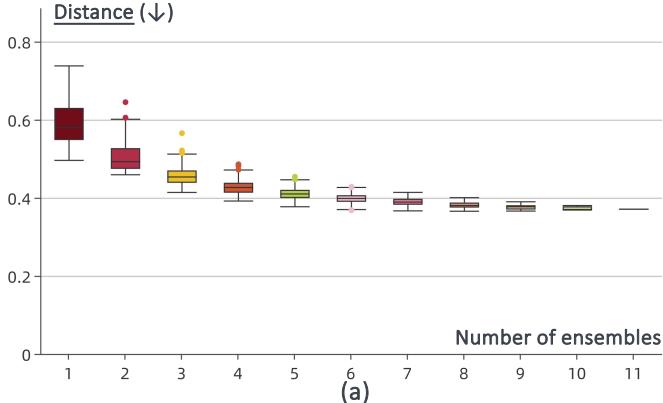


Figure S4. Ablation studies of the robust ensemble. (a) highlights a consistent improvement in performance with an increase in ensemble size, together with the decrease in the differences among various ensemble strategies. (b) shows how the robust ensemble significantly improves human alignment across all dimensions.

E. Additional Results

In this section, we will provide additional results which do not fit in the original paper.

E.1. Ranking

While the results of our method are still largely limited by API request limits, we would like to show some preliminary results for how existing text-to-3D generative models perform according to our methods in different criteria. Table R2 shows the top four methods according to GPT-4V in all six criteria. We can see that MVDream [60] is ranked the first across various criteria. Latent-NeRF [41] is able to achieve strong performance, ranked second or third across all criteria. Similarly, Prolific dreamer [68] achieves comparable performance as latent-NeRF except for Plausibility. Finally, Instant3D [36] is ranked the fourth places in all but Plausibility. Instant3D is ranked the third place in Plausibility, above Prolific dreamer. Dreamfusion [51] is able to achieve good Plausibility, probably thanks to its geometry regularization.

E.2. GPT-4V Comparison Examples

In Figure S6, we show some more examples of GPT-4V’s analysis and answers to this task (using “left / right / equal” instead of “1 / 2 / 3” for better clarity). These examples demonstrate how the evaluator is able to make comparisons of the paired visual input based on close observations and detailed analysis.

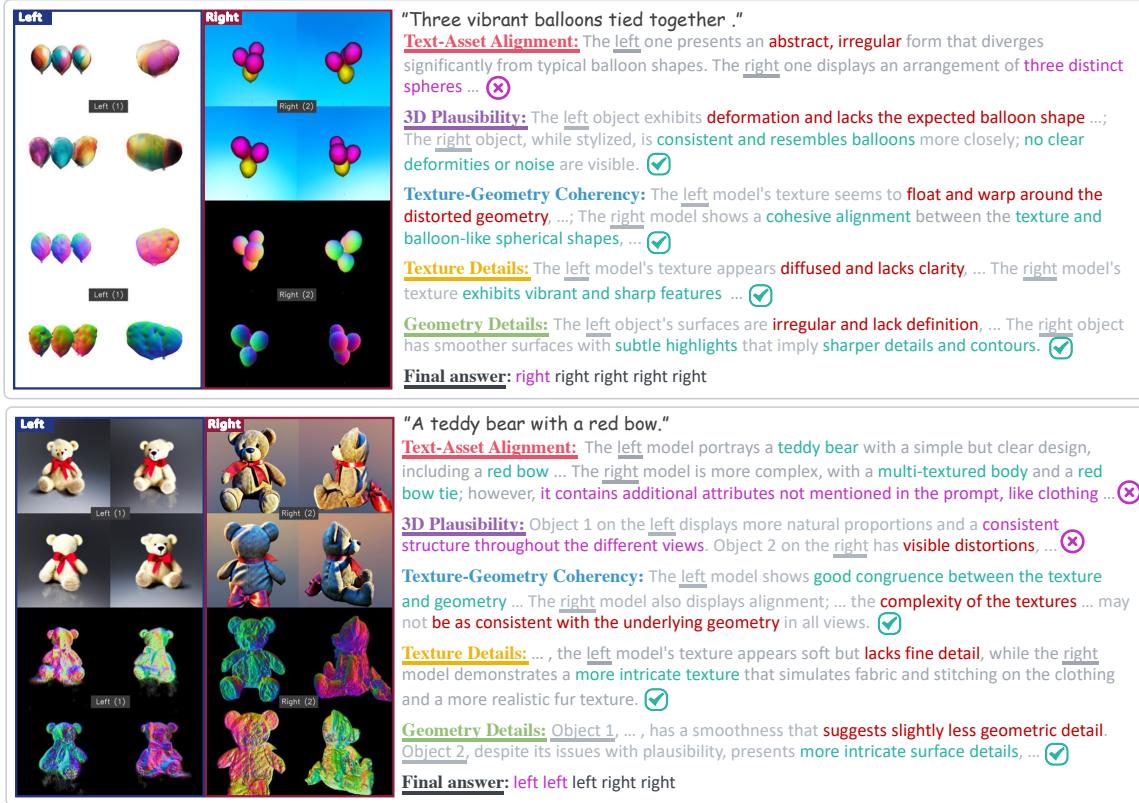
F. Failure cases

We present some typical failure cases in Figure S5. In the first instance, GPT-4V fails to detect the mismatch in the number of balloons between the two objects, thereby highlighting its limitations in accurately counting specific

quantities. In the second instance, both objects exhibit significant issues with their underlying geometries. The object on the left presents severe 3D plausibility problems, as evidenced by the presence of multiple faces and leg. Both objects are plagued by low-level geometric noise. GPT-4V demonstrates sensitivity to low-level noise, consequently overlooking the overarching issue with the object on the left. Such problems can potentially be rectified if one can provide a larger number of views to GPT-4V.

Table R2. Top-4 methods for different criteria according to our metrics.

	Alignment	Plausibility	Color-Geo	Texture	Geometry
1st	MVDream	MVDream	MVDream	MVDream	MVDream
2nd	Prolific Dreamer	Latent-NeRF	Latent-NeRF	Prolific Dreamer	Prolific Dreamer
3rd	Latent-NeRF	Instant3D	Prolific Dreamer	Latent-NeRF	Latent-NeRF
4th	Instant3D	Dreamfusion	Instant3D	Instant3D	Instant3D



✖ Wrong

✓ Correct

Figure S5. **Failure cases.** The analysis of GPT-4V can sometimes be *partially* wrong. We show several typical examples for the “Text-Asset Alignment” and “3D Plausibility” criteria.

Left

Right

"A small, rustic cabin sits alone in a peaceful, snow-covered forest."

Text-Asset Alignment: The left one ... a cabin ... **aligning with the "snow-covered forest"** description. The right one ... a cabin with an **ambiguous environment** and **less visible snow**.

3D Plausibility: The left one .. a clear cabin with **discernible roof and walls**, appearing **more cohesive overall**; The right one has parts that are **less defined**, with some **areas looking melted or incomplete**.

Texture-Geometry Coherency: The texture of the left cabin **aligns well with the underlying geometry** ... ; The right cabin ... **not as precise as the left cabin**.

Texture Details: The left cabin has a **sharper, more detailed texture with visible wood grains** and a **snow** effect that looks realistic. The right cabin has a **softer texture**, lacking the same level of crisp detail.

Geometry Details: The left one ... **more recognizable architectural elements** such as **windows, a door, and a chimney**; The right one has **less distinct features**, overall appearing smoother and lacking the same level of detail.

Final answer: left left left left left

Left

Right

"A lamp casting a warm glow."

Text-Asset Alignment: The left object is a table lamp with a colorful lampshade and an **irregularly shaped stand** ... The right ... that **accurately adheres to "casting a warm glow"**.

3D Plausibility: Object 1 appears to have a **more plausible structure with fewer artifacts** than Object 2. Object 2 demonstrates **several protrusions and disjointed elements** that detract from its plausibility as a lamp.

Texture-Geometry Coherency: The left object shows **some inconsistencies**, especially **at the base**. The right one has a **coherent alignment** between its texture and geometry ...

Texture Details: The textures on the left object ... are **less sharp** and **do not convey a clear material quality**. The texture of the right object appears **more detailed and realistic**, with materials that **reflect light akin to metal and glass**,

Geometry Details: Object 2 exhibits a **higher level of complexity and detail** in the geometry of its components, suggesting sharper detail, especially around the lantern's details and the **ornamental aspects**.

Final answer: right left right right right

Left

Right

"A crying sofa."

Text-Asset Alignment: The left one ... **do not succinctly convey** the concept of a "crying sofa". The right one ... but **also does not clearly represent** the idea of "crying". Neither model has attributes that explicitly suggest crying.

3D Plausibility: The left ... **a high degree of warping and distortion**, with elements that are **malformed and far from resembling** a stable structure. The right object ... **maintains a more consistent shape and structure** ...

Texture-Geometry Coherency: The left one appears ... with **no discernible patterns** that relate to a typical sofa texture. The right one has a **texture that aligns with its geometry** ...

Texture Details: The left one ... **lacking any realistic sofa material qualities**. The right one presents a texture with **a degree of realism**, depicting shadows and highlights consistent with the geometry of cushions and armrests.

Geometry Details: The left object's geometry is **extremely warped**, being **difficult to discern any clear details**. The right object has a **more defined structure**, ...

Final answer: equal right right right right

Left

Right

"A compact, cylindrical, vintage pepper mill, with a polished, ornate brass body, slightly worn from use, placed beside a porcelain plate on a checkered tablecloth."

3D Plausibility: Both models seem **solid and plausible**. The left one has a **decorative complexity that is coherent**, while the right one, although simpler, **also represents a consistent form**.

Texture Details: The left model shows a **high level of detail** ... effectively conveys the **porcelain and brass textures suggested in the prompt**. The right model has a **more subdued texture appearance**, ...

Final answer: equal right

Left

Right

"A chair made from polished oak."

3D Plausibility: The left model shows **implausible distortions and uneven surfaces**, while the right model has a **consistent structure** that is plausible as a chair.

Geometry Details: The normal maps of the left ... **chaotic arrangement of colors** with little to no discernible **consistent pattern or structure**. The right ... **more orderly and structured appearance**, with **more defined shapes and patterns**.

Final answer: right right

Figure S6. Additional examples of the analysis by GPT-4V.



Please help us to create a set of text prompts for text to 3D generation. Now, I will give you some more tips and requirements for the prompt creation. First, we will decide the complexity and creativity:

1. Complexity:

- Structure Complexity: Simplicity or intricacy of the object's design
- Number of Elements: Single or multiple subjects
- Relationships: Interactions and relations between multiple objects
- Environmental Complexity: Simplicity or intricacy of the scene and surroundings
- Detail Level: From simple shapes to complex structures with fine details

2. Creativity:

- Novelty: Common, everyday items to unique, imaginative creations
- Conceptual Innovation: Degree of abstract or unconventional thinking required
- Emotional Impact: Potential to evoke emotions or thoughts
- Storytelling: Inclusion of elements that suggest a backstory or narrative

Upon determining these two aspects, we then turn our attention to the content of the prompt. Initially, a subject selection is conducted, followed by the decision on further constraints or descriptions premised on this chosen subject:

1. Subject or Object Categories:

- Living Beings: Humans, animals, mythical creatures, insects, aquatic life
- Plants: Trees, flowers, bushes, grass, aquatic plants
- Buildings and Structures: Houses, skyscrapers, bridges, tunnels, castles
- Vehicles: Cars, bikes, planes, boats, spacecraft
- Furniture: Chairs, tables, sofas, shelves, beds
- Electronics: Phones, computers, cameras, kitchen appliances
- Household Items: Utensils, decor, tools, containers
- Clothing and Accessories: Dresses, shoes, jewelry, hats, bags
- Abstract Objects: Geometric shapes, artistic sculptures
- Natural Elements: Fire, water, rocks, clouds
- Food and Beverages: Fruits, vegetables, dishes, drinks

2. Geometry Constraint:

- Volume: Specify if the object is hollow, solid, porous, or layered.
- Symmetry: Define if the object should be symmetrical, asymmetrical, or radially symmetrical.
- Contours: Indicate if the contours should be smooth, jagged, irregular, or undulating.
- Internal Structures: State if the inside should be empty, compartmentalized, or multi-layered.
- Shape: Mention specific geometric shapes that should be part of or inspire the object, e.g., cone, cylinder, sphere.

3. Appearance:

- Colors: Specific colors, patterns, gradients
- Materials: Wood, metal, glass, fabric, stone
- Textures: Smooth, rough, furry, scaly
- Finish: Glossy, matte, translucent, opaque
- Size: Small, medium, large, specific dimensions
- State: New, old, worn, pristine

4. Status/Motion:

- Static: Still, motionless
- Dynamic: Moving, changing
- Emotional State: Happy, sad, angry (for living beings or anthropomorphized objects)
- Physical State: Broken, intact, in use
- Interaction: Interacting with another object or environment

5. Styles:

- Aesthetic: Minimalistic, ornate, modern, vintage
- Cultural: Asian, African, Western, Middle Eastern
- Emotional: Cheerful, gloomy, energetic, calm
- Functional: Practical, decorative, symbolic
- Conceptual: Abstract, realistic, surrealistic, impressionistic

6. Scene and Surroundings:

- Environment: Indoor, outdoor, urban, rural, natural, fantastical
- Context: Part of a larger scene, event, or story
- Lighting: Day, night, artificial, natural, shadows, highlights
- Weather: Sunny, rainy, cloudy, stormy (for outdoor scenes)
- Interactions: How the subject interacts with its surroundings
- Scale: The relative size of the object in the scene

Try to understand the requirement first, before I provide some more information.



Understood. Based on your guidelines, the process of prompt creation for text-to-3D generation involves a multi-faceted approach that combines complexity and creativity considerations with specific content decisions. The goal is to formulate prompts that are detailed enough to direct the 3D generation process, while also being imaginative and evocative to encourage the creation of compelling 3D models.

Complexity considerations help in determining the structural and environmental intricacy, the number of elements and their relationships, as well as the overall detail level. **Creativity** involves novelty, conceptual innovation, emotional impact, and the inclusion of storytelling elements.

When crafting prompts, one should choose a subject or object category and then refine the prompt with additional constraints or descriptions, such as geometry, appearance, status/motion, style, and scene specifics. Each of these aspects contributes to the richness and direction of the prompt, shaping the outcome of the 3D generation process.

It's clear that prompts can range from very simple directives (like "A silver spoon") to highly complex and storied scenarios ("A timeworn leather journal with cryptic inscriptions lying on the altar of an abandoned temple, under the eerie glow of the full moon").

With these frameworks in mind, you can create prompts that are not only technically precise but also deeply imaginative, leading to a diverse portfolio of 3D objects and scenes. Now, I'm ready for any further information or the next steps you have regarding the prompt creation.

Figure S7. Examples of guiding GPT for automatic prompt generation. (Part 1/2)



I will first show you some examples:

"a bald eagle carved out of wood",
"a banana peeling itself",
"a beagle in a detective's outfit",
"a beautiful dress made out of fruit on a mannequin. Studio lighting high quality high resolution",
"a beautiful dress made out of garbage bags on a mannequin. Studio lighting high quality high resolution",
"a cute steampunk elephant",
"a dachshund dressed up in a hotdog costume",
"a delicious hamburger",
"a dragon-cat hybrid",
"a DSLR photo of a baby dragon drinking boba",
"a DSLR photo of a baby dragon hatching out of a stone egg",
"a DSLR photo of a baby grand piano viewed from far away",
"a DSLR photo of a bagel filled with cream cheese and lox",
"a DSLR photo of a bald eagle",
"a DSLR photo of a barbecue grill cooking sausages and burger patties",
...



Thank you for the examples. I've reviewed the styles. How can I assist you next?



Please create 50 common prompts, each within 10 words, with a diversity of complexity.



Certainly! Here are 50 diverse prompts that vary in complexity.

1. A sunflower field at sunset.
2. A pair of worn hiking boots.
3. A cat sleeping in a sunny window.
4. A busy urban street scene.
5. A portrait of a woman with green eyes.
6. A sushi platter with a variety of rolls.
7. An old library with tall bookshelves.
8. A vintage convertible on a coastal road.
9. A snowman wearing a top hat and scarf.
10. A child's first bicycle ride.
11. A bowl of fresh summer berries.
12. A cozy cabin in the snow.
13. A jazz band playing in a smoky bar.
14. A knight's armor on display.
15. A traditional Venetian mask.
16. An antique pocket watch with intricate engravings.
17. A colorful hot air balloon in flight.
18. A chef preparing a flambé dish.
19. A mermaid lounging on a rock.
20. A medieval castle at dawn.
21. ...



Please create 50 prompts about living creatures, with a diversity of creativity.



Common Living Creature Prompts:

1. Sleeping tabby cat curled in a sunbeam.
2. Goldfish swimming in a round bowl.
3. Green tree frog clinging to a rain-soaked window.
4. Orange monarch butterfly resting on a dandelion.
5. Spotted ladybug crawling on a green leaf.
6. White-tailed deer standing in a meadow at dawn.
7. Clownfish peeking out from sea anemone tentacles.
8. Brown rabbit nibbling on a clover leaf.
9. Black and white dairy cow in a grassy field.
10. Bumblebee landing on a sunflower's center.
11. Red fox prowling through a snowy forest.
12. ...

Unusual and Creative Living Creature Prompts:

1. Chameleon with color-changing spots forming digital numbers.
2. Caterpillar with a keyboard pattern on its back.
3. Swan with feathers resembling soft, white origami folds.
4. Octopus with transparent tentacles revealing gears inside.
5. Gecko with suction pads that glow like neon lights.
6. Flamingo balancing on a sphere instead of standing in water.
7. Frog with a translucent skin displaying a mechanical heart beating.
8. Koala with a patchwork of velvety moss instead of fur.
9. Seahorse with a carousel-inspired saddle and fin designs.
10. Tortoise with a shell that looks like stained glass.
11. Jellyfish with bioluminescent tentacles shaped like lightning bolts.
12. ...

Figure S8. Examples of guiding GPT for automatic prompt generation. (Part 2/2)