

Benchmarking and Learning Multi-Dimensional Quality Evaluator for Text-to-3D Generation

Supplementary Material

A. Overview

This supplementary material includes additional details on benchmark construction, evaluator implementation, and experiment results, which cannot be fully covered in the main paper due to limited space. In Appendix B, we first introduce more details about the benchmark construction, including the definition of eight prompt categories and the instruction used in GPT-4. Then, we present more implementation details about the proposed evaluator, including network details and training strategy in Appendix C. Finally, we provide additional experimental results to demonstrate the effectiveness of our metric in Appendix D.

B. More Details on Benchmark Construction

B.1. Definition of Different Prompt Categories

Considering the number of objects and their complexity, creativity, and relationship, we define eight prompt categories in our benchmark. The definitions of these categories are reported as follows:

- *Basic*. Descriptions about a single object without detailed geometry or appearance details, where some global properties (e.g., color, shape, material) could be included. Examples: “A green apple”, “A square table”, and “A wooden chair”.
- *Refined*. Descriptions about a single object with one simple specification of the geometry or appearance details. Examples: “A yellow rubber duck has only one foot”, “A plastic cup with a printed logo”, and “An imperial state crown of England”.
- *Complex*. Descriptions about a single object with more than two detailed specifications of geometry or appearance details. Examples: “A brown teddy bear, fur matted, one eye missing”, “A turtle standing on its hind legs, wearing a top hat and holding a cane”, and “An oversized, porous, sphere-shaped birdcage, made of woven golden wires”.
- *Fantastical*. Descriptions about a single object with high creativities that are not, or are generally unlikely to be, found in the real world. Examples: “A frog with a translucent skin displaying a mechanical heart beating”, “A white cat has three tails, made of white, purple and black crystals”, and “A tiger dressed as a doctor”.
- *Grouped*. Descriptions about multiple objects without interactions, where some global properties (e.g., color, shape, material) could be included. Examples: “A delicious hamburger and a green apple”, “A red pig and a

huge drum”, and “A round table and a square chair”.

- *Action*. Descriptions about multiple objects with interaction about action relations, where some geometry or appearance details could be included. Examples: “A humanoid robot with a top hat is playing the cello”, “A black cat with white feet is sleeping peacefully beside a carved pumpkin”, and “A dog is eating a red apple with its tail raised”.
- *Spatial*. Descriptions about multiple objects with interactions about spatial relations, where some geometry or appearance details could be included. Examples: “A red apple on a round ceramic plate”, “A strong football player wearing a number seven jersey next to a blue soccer”, and “A blue rose in a crystal, symmetrical vase”.
- *Imaginative*. Descriptions about multiple objects with interactions, where objects or interactions are not or are generally unlikely to be found in the real world. Examples: “A panda with a wizard hat is reading a newspaper”, “A ghost is eating a golden apple”, and “A raccoon astronaut is holding his helmet”.

B.2. Prompt Generation

To ensure the comprehensiveness and diversity of prompts generated by GPT-4 [11], we define four aspects to consider during the prompt generation: object categories, geometry properties, appearance properties, and object interactions. These aspects are detailed as follows:

- **Object Categories**. To ensure that the generated prompts encompass the most common object classes relevant to text-to-3D tasks, we first establish clear definitions for object categories, such as “live beings”, “animals”, “plants” and so on. This approach ensures prompt diversity and avoids the generated prompts being limited to a single category.
- **Geometry Properties**. To generate more vivid and realistic 3D objects, it is essential to include detailed descriptions of geometry properties in prompts. Objects can be characterized based on their volume, shape, or size, including terms such as “symmetrical”, “cylinder”, and “small”.
- **Appearance Properties**. Similar to geometry properties, appearance properties are essential for prompt generation. Common attributes include colors, which are frequently used to describe objects’ appearance, and textures, which capture surface characteristics such as “smooth”, “rough”, and “furry”. Additionally, materials, such as “metal”, “glass”, and “fabric”, also play a signifi-



Figure S1. Training samples used in our subjective experiment.

cant role in defining appearance.

- **Object Interactions.** Object interactions can be classified into two types: spatial and action relationships. Spatial relationships utilize terms such as “on” and “below” to accurately define the positional relationships between objects. Action relationships describe the actions or behaviors of objects, including verbs like “wear”, “watch”, and “hold”. By integrating these words with different object categories, GPT-4 can generate prompts encompassing various interactions.

After defining eight prompt categories and four aspects, we design a universal template to provide GPT-4 with essential context for the prompt generation task. Detailed instructions are shown in Fig. S8 and Fig. S9. Following these instructions, GPT-4 can gain a foundational understanding of the task and generate a list of prompts based on evaluator input. By editing prompt categories and lengths, GPT-4 can efficiently produce prompts across different categories.

B.3. Mesh Generation and Visualization

In recent years, significant advancements have been made in 3D generative methods. In this paper, we employ DreamFusion [12], Magic3D [7], SJC [15], TextMesh [14], 3DTopia [4], Consistent3D [17], LatentNeRF [10], and One-2-3-45++ [8] to generate 3D textured meshes. Here, we provide a brief introduction of each method.



Figure S2. Illustration of the environment platform.

- **DreamFusion** pioneers the paradigm of optimizing a unique 3D representation per text input or image, guided by powerful pre-trained 2D diffusion models.
- **Magic3D** introduces a coarse-to-fine optimization strategy with two stages. This approach improves both speed and quality.
- **SJC** converts a pre-trained 2D diffusion generative model on images into a 3D generative model of radiance fields, without requiring access to any 3D data.
- **TextMesh** employs a novel way to fine-tune the mesh texture, removing the effect of high saturation and improving the details of the output textured mesh.
- **3DTopia** is a two-stage text-to-3D generation method. The first stage uses a diffusion model to quickly generate candidates. The second stage refines the assets chosen from the first stage. It can generate high-quality general 3D assets within 5 minutes using hybrid diffusion priors.
- **Consistent3D** uses deterministic sampling priority to ensure that different generated results are visually more consistent and have higher details when generating 3D objects. Compared to traditional random sampling methods, this approach reduces variability in generated results, ensuring that the produced 3D objects align more consistently with the specified text prompt.
- **LatentNeRF** guides the 3D generation process by encoding shape information in latent space. In this way, it can more effectively generate high-quality textures and shapes that match specific shapes while preserving details.
- **One-2-3-45++** advances multi-view 3D generation via an enhanced Zero123 [9] module enabling simultaneous cross-view attention, alongside a multi-view conditioned 3D diffusion module performing coarse-to-fine textured mesh prediction over time.

We present additional generated samples along with the evaluation scores for each method in Fig. S10.

B.4. Subjective Experiment Procedure

Training Session. Before the subjective experiment, we scale the generated meshes proportionally to ensure that

Table S1. Comparison of the existing text-to-3D datasets. ‘ \times ’ represents the scores are not available.

Benchmark	Prompt Categories	Number of Generative Methods	Rating Dimensions	Number of Annotated Samples	Number of Rating Score	Annotation Type
T ³ Bench [3]	Single Object, Single Object with Surroundings, Multiple Objects	7	Alignment, Quality	630	$630 \times 2 \times \text{unknown}$	\times (Absolute Score)
GPTEval3D [16]	Creativity, Complexity	13	Text-Asset Alignment, 3D Plausibility, Texture details, Geometry details, Texture-geometry coherency	234 pairs	$234 \times 5 \times 3$	\times (Preference Score)
MATE-3D	Basic, Refined, Complex, Fantastical, Grouped, Action, Spatial, Imaginative	8	Alignment, Geometry, Texture, Overall	1,280	$1,280 \times 4 \times 21$	\checkmark (Absolute Score)

Table S2. Performance comparison with evaluators in GPTEval3D and T³Bench.

KRCC	Alignment	Geometry	Texture	Overall
HyperScore	0.517	0.572	0.622	0.603
GPTEval3D	0.355	0.356	0.432	0.391
SRCC	Alignment	Geometry	Texture	Overall
HyperScore	0.739	0.782	0.811	0.792
T ³ Bench-Quality	0.619	0.497	0.494	0.540
T ³ Bench-Alignment	0.407	0.366	0.376	0.383

each mesh fits within a cube defined by the range $[-1, 1]$. To enhance the reliability of the subjective scores, we use 10 samples whose corresponding prompts are excluded from MATE-3D to train subjects, helping them understand the rating rules. The training samples are selected to represent a comprehensive quality range, allowing viewers to understand the dataset comprehensively. We assign reference scores to the training samples and present the samples twice, requiring viewers to score them during the second viewing. If the scores given by the viewers show a high correlation with the reference scores, we conclude that they have grasped the rating principles and can provide reliable scores. Conversely, if viewers assign biased scores from the references, we repeat the training procedure until they provide reasonable results. The 10 training samples are illustrated in Fig. S1.

Experimental Environment. To enable subjects to observe 3D objects from various viewpoints and provide more accurate ratings, we employ an interactive approach for the experiment. Our interactive renderer is developed as a web application utilizing the Three.js library [6]. To effectively capture the scene, we employ an orthographic camera, with its field of view defined by the camera frustum. Subjects can control the camera’s orientation through mouse movements, which enables them to adjust their viewpoints dynamically. To minimize the influence of the background on the subjects’ evaluation, we set the background color to gray, specifically implemented using the code

`scene.background = newTHREE.Color(0xaaaaaa)`. The platform, as illustrated in Fig. S2, allows subjects to navigate between previous and next samples and rate scores from four dimensions. The subjective experiment is conducted on 27-inch AOC Q2790PQ monitors with a resolution of 2560×1440 in an indoor laboratory environment under standard lighting conditions.

Outlier Detection. In total, we generate 1,280 textured meshes from 160 prompts with eight generative methods. To mitigate visual fatigue associated with prolonged experiment durations, we randomly divide the 1,280 samples into 10 sessions. To detect outliers during the subjective experiment, each rating session includes one extremely low-quality sample and one duplicate sample as “trapping samples”. Consequently, each session comprises 130 samples. After collecting the subjective scores, we implement two consecutive steps to identify outliers from the raw data. First, we identify outliers based on the “trapping samples” results. If a subject provides a high score that exceeds the expected threshold for the extremely low-quality sample or provides significantly different scores for the two duplicate samples, we exclude the raw scores of the subject from our analysis. Second, we apply the outlier detection method described in ITU-R BT.500 [13] to conduct a further examination, removing any additional outliers identified through this process. As a result, four outliers are identified and eliminated from the subjective scores. Finally, we collect 17 scores for each sample.

B.5. Comparison with Other Benchmarks

Previous works [3, 16] have presented some text-to-3D benchmarks. We compare MATE-3D with the existing benchmarks from multiple perspectives in Tab. S1. Note that for the column of “Number of Annotated Samples”, T³Bench and MATE-3D provide “Absolute Score” for each sample, whereas GPTEval3D creates pairs of two samples and assigns ranking orders, referred to as “Preference Score” within each pair. The column of “Number of Rating Scores” is calculated as “Number of Annotated Sam-

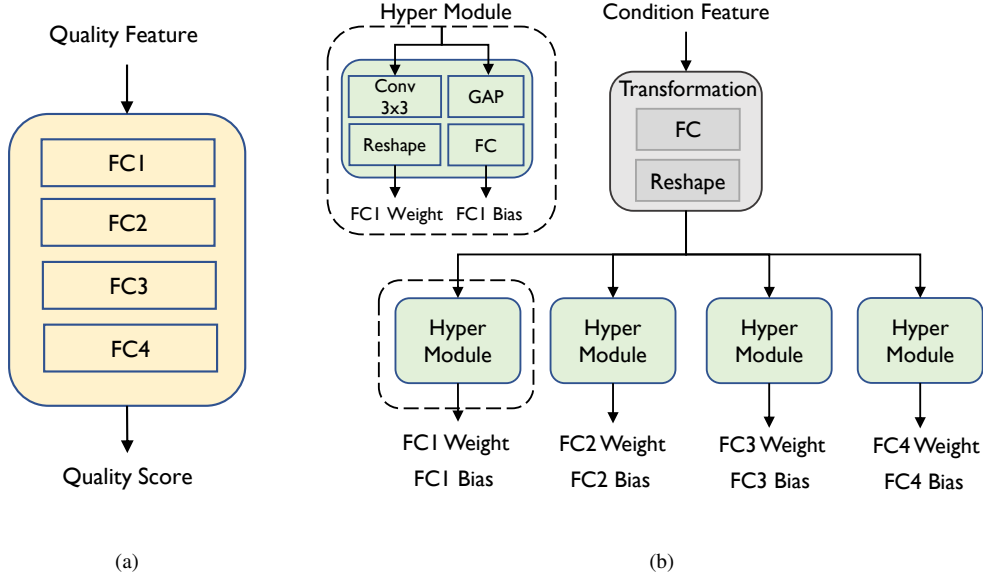


Figure S3. (a) Architecture of the mapping head; (b) Architecture of the hypernetwork.

ples” \times “Number of Rating Dimensions” \times “Subject Number for Each Sample”. Additionally, since the scores for T^3 Bench and GPTEval3D are not publicly available, we cannot evaluate the performance of objective metrics on these two benchmarks.

From Tab. S1, it is clear that our benchmark presents several noticeable advantages. First, we incorporate a broader range of prompt categories, ensuring diversity and representativeness in the prompt generation. Second, we introduce four rating dimensions, offering a more comprehensive assessment of sample quality compared to T^3 Bench. Although GPTEval3D employs more dimensions, it only provides preferences for a limited number of pairs. Third, to promote robust and unbiased scoring, we annotate all generated samples and recruit 21 subjects to rate 1,280 samples across four dimensions. Additionally, we apply comprehensive outlier detection to refine the MOS, reducing the impact of anomalous ratings. We will make the MOS publicly accessible, facilitating further validation and experimentation by other researchers.

We further compare the evaluators proposed in GPTEval3D and T3Bench with HyperScore. i) **Comparison with GPTEval3D**: The evaluator GPTEval3D is based on GPT-4v, which can only rank generated results from the same prompt in terms of five dimensions. However, directly ranking two samples in these methods can only know which one is better, but not how much better. In comparison, the proposed evaluator provides absolute scores for different samples, which can not only compare samples generated from different prompts but also know how much the quality dif-

ference between the two samples is. We report the KRCC results of HyperScore and GPTEval3D on MATE-3D in the first table of Tab. S2, we can see HyperScore perform better. ii) **Comparison with T^3 Bench**: T^3 Bench proposes two separate evaluators for quality and alignment measurement, which cannot handle multi-dimensional quality assessment. We report the SRCC results of HyperScore and two evaluators in T^3 Bench on MATE-3D in the second table of Tab. S2, we can see HyperScore can evaluate different dimensions more effectively than two evaluators in T^3 Bench.

C. More Details on Evaluator Implementation

C.1. Network Details

Utilization of Pre-trained Model. The used visual encoder in HyperScore is Vision Transformer [2] with 16×16 patch embeddings (namely ViT-B/16) in CLIP-Visual. The textual encoder is also the pre-trained transformer in CLIP-Textual. The visual, textual, and condition features all have a size of $D = 512$ while the quality feature has a size of $D_q = 224$.

Architecture of HyperNetwork and Mapping Head. In the evaluator, we use a hypernetwork $\pi(\cdot)$ to generate the parameter weights for a mapping head $\psi(\cdot)$. We illustrate the network structure of $\pi(\cdot)$ and $\psi(\cdot)$ in Fig. S3. Seeing Fig. S3a, the mapping head contains four fully connected (denoted by FC1-FC4) layers and takes the quality feature as input to generate the quality score. The input sizes of four FC layers are 224, 112, 56, and 28 as shown in Tab. S3. To inject the dimension-related information into the mapping

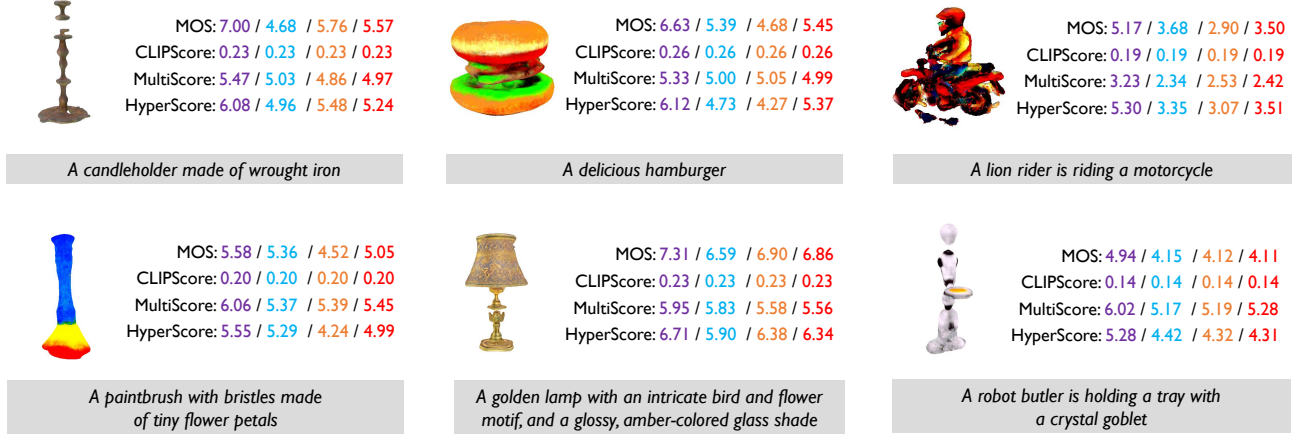


Figure S4. Additional exemplary samples with their MOSs and the predicted scores of different metrics. The four scores in each row denote alignment, geometry, texture, and overall quality, respectively.

head, we use the hypernetwork shown in Fig. S3b to generate the weights and biases for FC layers. More specifically, the condition feature after transformation is fed into four independent hyper modules, and each hyper module will output the weight and bias for the corresponding FC layer in the mapping head. We also declare the feature size for the weight and bias generated of the FC1 layer in Tab. S3.

C.2. Loss Function

Given K evaluation dimensions, we finally obtain $\{\hat{q}_i\}_{i=1}^K$ as predictions. Denoting the subjective MOS as $\{q_i\}_{i=1}^K$, we first define a regression loss as:

$$L_{reg} = \frac{1}{KB} \sum_{b=1}^B \sum_{i=1}^K (\hat{q}_i^b - q_i^b)^2, \quad (1)$$

where the superscript “ b ” denotes the b -th samples in a mini-batch with the size of B .

Considering that different evaluation dimensions have various focuses, to avoid learning homogeneous features, we define a feature disentangling loss for the condition features $\{f_c^i\}_{i=1}^K$:

$$L_{dis} = \frac{2}{K(K-1)} \sum_{i \neq j} \max(\epsilon, \cos(f_c^i, f_c^j)), \quad (2)$$

where $\cos(\cdot)$ denotes the cosine similarity between two features; ϵ represents the margin that controls the divergence among features and we set $\epsilon = 0$ in the implementation. By introducing L_{dis} , we can increase the discrepancy among different evaluation dimensions during the network training.

Finally, the overall loss function for training is defined as:

Table S3. The feature size of each module in the hypernetwork and mapping head.

Module	Layer	Input Size	Output Size
Mapping Head	FC1 Layer	224	112
	FC2 Layer	112	56
	FC3 Layer	56	28
	FC4 Layer	28	1
Transformation	FC Layer	512	5,488
	Reshape	5,488	$112 \times 7 \times 7$
FC1 Weight Generation	3×3 Conv	$112 \times 7 \times 7$	$512 \times 7 \times 7$
	Reshape	$512 \times 7 \times 7$	224×112
FC1 Bias Generation	Global Avg Pool	$112 \times 7 \times 7$	$112 \times 1 \times 1$
	FC Layer	112	112

$$L = L_{reg} + \lambda L_{dis}, \quad (3)$$

where λ is the weighting factor, we simply set $\lambda = 1$.

C.3. Training Strategy

We train the proposed HyperScore on MATE-3D for 30 epochs with a batch size of 8. During the training and testing process, all rendered images are resized into the resolution of 224×224 . We use the Adam [5] optimizer with weight decay $1e-4$. The learning rate is set separately as $2e-6$ and $2e-4$ for the pre-trained visual encoder and other parts (note that the textual encoder is frozen) and is reduced by a rate of 0.9 every 5 epochs.

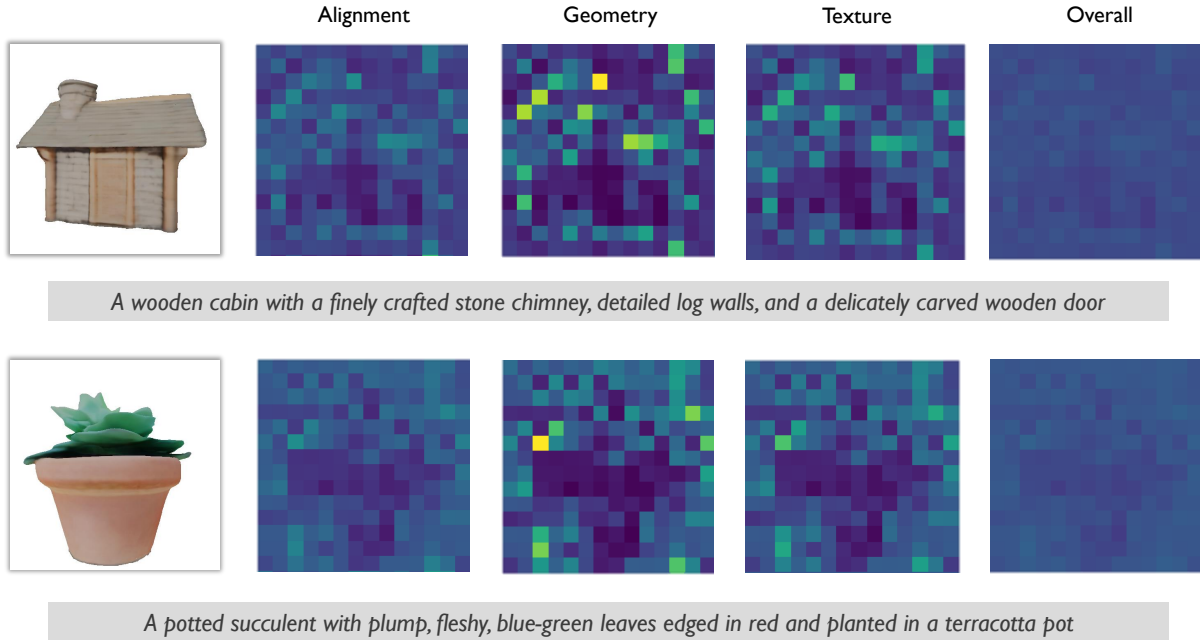


Figure S5. Visualization for the weight maps used for conditional feature fusion.

D. Additional Experimental Results

D.1. Visualization

We provide additional qualitative samples in Fig. S4 for visualization. From the figure, we can observe that HyperScore presents more accurate predictions. In comparison, MultiScore usually assigns similar scores across different dimensions, making it challenging for users to determine which factor impacts visual perception more.

Furthermore, we visualize the weight maps used for conditional feature fusion in Fig. S5. From the figure, we can observe that the weight maps differ across evaluation dimensions, demonstrating that the condition features contribute to distinguishing these dimensions. Meanwhile, we notice that the weight maps for the geometry evaluation have larger element values at the edges of objects, which may help to seize the shapes better. In comparison, the overall evaluation exhibits a more uniform distribution of the weight maps, probably because it needs to consider various factors during the evaluation process.

To validate the evaluation ability for ground truth data, we download 2000 meshes from Objaverse [1] and use HyperScore to access their quality. The average scores for four dimensions are [7.9, 7.6, 6.7, 7.4], which is generally higher than the 3D generation quality. Fig. S6 illustrates two evaluated examples. There also exist relatively low-quality samples in Objaverse such as “horse”.

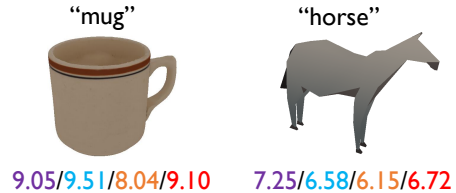


Figure S6. Evaluation results for examples in Objaverse.

D.2. Additional Ablation Studies

Ablation Study for the Aggregation Strategy. We test the performance of HyperScore with different aggregation strategies between the visual and textual features. Except for the used element-wise multiplication (denoted by \odot), we choose two other strategies, *i.e.*, addition (denoted by $+$) and concatenation (denoted by \oplus). According to the results in Tab. S4, we can see that the element-wise multiplication performs best on all evaluation dimensions, and the addition and concatenation both provide relatively inferior performance, which justifies our choice for the aggregation strategy.

Ablation Study for the Viewpoint Count. To perform evaluation, we render textured meshes into $M = 6$ images from six perpendicular viewpoints (*i.e.*, along the positive and negative directions of the x, y, and z axes). We further test the performance of HyperScore under different M values to investigate the influence of the number of view-

Table S4. Ablation study for the aggregation strategy between visual and textual features. Results of SRCC are reported.

Aggregation	Alignment	Geometry	Texture	Overall
$f_{v,c}^i + f_t^{eot}$	0.708	0.769	0.783	0.767
$f_{v,c}^i \oplus f_t^{eot}$	0.717	0.766	0.789	0.767
$f_{v,c}^i \odot f_t^{eot}$	0.739	0.782	0.811	0.792

points and report the results in Tab. S5. Note that the corresponding camera locations of different M are declared in Tab. S6. From Tab. S5, it can be observed that as the M value increases, the performance initially improves and then decreases. The reason may be that increasing M provides more information for a better prediction when M is a small number. However, when M becomes large, the increase in M can affect performance to some extent due to information redundancy. Meanwhile, a larger M also incurs higher computational complexity. Therefore, to achieve the balance between performance and complexity, we consider $M = 6$ as a suitable choice.

Ablation Study for the Prompt Design. In our implementation, to obtain multiple condition features, we first transform the meta texts (i.e., “alignment quality”, “geometry quality”, “texture quality”, “overall quality”) into text tokens and then insert them into the front of K learnable prompts with $L = 12$ tokens. Here we further explore the impact of L on the performance and illustrate the results in Fig. S7. From the figure, we see that HyperScore performs the best with $L = 12$. Short prompts may constrain the learning space, potentially leading to the loss of crucial information. Conversely, too long prompts might introduce unnecessary noise, which may obscure important information and degrade performance. Therefore, balancing the prompt length is critical to optimizing the metric performance.

We further replace these learnable prompts with fixed tags and test the performance of the case. Specifically, we define the tags for the four quality dimensions as:

- **Alignment:** “quantity, attributes, position, location”;
- **Geometry:** “shape, size, hole, edge, surface”;
- **Texture:** “color, material, clarity, texture, contrast”;
- **Overall:** “quantity, attributes, position, location, shape, size, hole, edge, surface, color, material, clarity, texture, contrast”.

The performance of the fixed tags is reported in Tab. S7, where we also report the performance of only using learnable tokens without the meta texts (denoted by “Learnable w/o meta”). We can observe that the learnable prompts with the meta texts outperform the fixed tags and alleviate the need for meticulous prompt design. Meanwhile, introducing the meta texts also benefits network optimization.

Ablation Study for the Loss Function. The proposed

Table S5. Ablation study for the number of rendered viewpoints. Results of SRCC are reported.

M	Alignment	Geometry	Texture	Overall
4	0.716	0.755	0.790	0.766
6	0.739	0.782	0.811	0.792
9	0.737	0.780	0.810	0.790
12	0.724	0.784	0.805	0.784
16	0.724	0.784	0.805	0.784

Table S6. Camera locations of different view counts.

M	Elevation Angle	Azimuth Angle
4	$-60^\circ, 60^\circ$	$0^\circ, 180^\circ$
9	$-60^\circ, 0^\circ, 60^\circ$	$0^\circ, 120^\circ, 240^\circ$
12	$-60^\circ, 0^\circ, 60^\circ$	$0^\circ, 90^\circ, 180^\circ, 270^\circ$
16	$-60^\circ, -30^\circ, 30^\circ, 60^\circ$	$0^\circ, 90^\circ, 180^\circ, 270^\circ$
6	$(0^\circ, 0^\circ), (0^\circ, 90^\circ), (0^\circ, 180^\circ), (0^\circ, 270^\circ), (90^\circ, 0^\circ), (-90^\circ, 0^\circ)$	

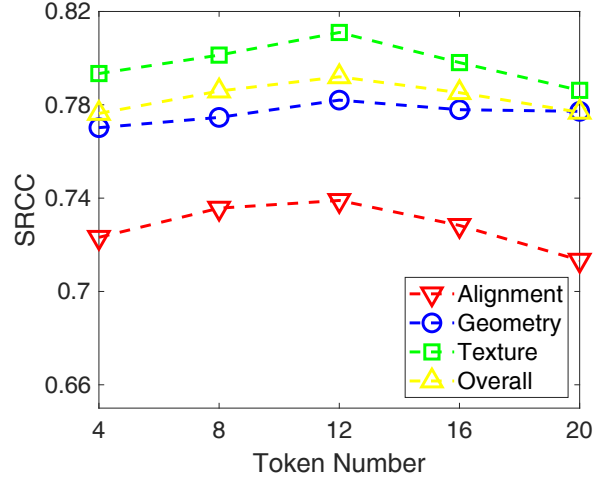


Figure S7. Performance comparison of the learnable prompt length.

network is trained using the regression loss L_{reg} , and the feature disentangling loss L_{dis} . We evaluate the effect of each loss function and report the results in Table S8. From the table, we can see that only utilizing L_{reg} can also achieve remarkable performance while L_{dis} benefits the evaluations of all dimensions. Considering L_{dis} only measures the linear similarity between features, one possible avenue for further improvement is to minimize non-linear dependence between two features, such as mutual information.

Ablation Study for the Input Image Type. The proposed network only uses texture rendering as input. We further test the impact of normal maps on model performance and reported the SRCC results in Tab. S9. It seems that feeding the normal maps into our model does not bring

Table S7. Ablation study for the prompt type. Results of SRCC are reported.

Prompt Type	Alignment	Geometry	Texture	Overall
Fixed	0.732	0.778	0.794	0.782
Learnable w/o meta	0.725	0.772	0.802	0.784
Learnable	0.739	0.782	0.811	0.792

Table S8. Ablation study for the loss function. Results of SRCC are reported.

Loss	Alignment	Geometry	Texture	Overall
L_{reg}	0.728	0.769	0.797	0.782
$L_{reg} + L_{dis}$	0.739	0.782	0.811	0.792

much improvement even for geometry evaluation, which is consistent with the conclusion in GPTEval3D. The reason for this may be that the backbones are trained on natural images rather than normal maps, so simply merging normal information impairs the performance.

Ablation Study for the Rotation Angle. A good 3D evaluation metric should be robust to changes of scale and viewpoint. HyperScore has scale invariance because the meshes are scaled into the unit ball before rendering. Meanwhile, HyperScore is robust to rotation because the training samples are in different poses. We report the performance under different azimuth angle in Tab. S10, where the results are relatively stable.

D.3. Performance Comparison on Different Prompt Categories

In this section, we test the performance of different metrics on eight prompt categories and report the results in Tab. S11. For the metrics that need fine-tuning, we perform a leave-one-category-out evaluation, that is, testing on one category while training on the other seven categories, which can validate the generalization performance for unknown scenes. From the table, we have the following observations: i) HyperScore achieves the best performance on the four dimensions of almost all prompt categories, demonstrating its sensitivity to fine-grained scenes. In contrast, although some metrics work well on partial categories (e.g., ImageReward on *Basic* and ViT+FT on *Spatial*), they may fail in other cases (e.g., ImageReward on *Complex* and ViT+FT on *Basic*). ii) Almost all metrics perform better when evaluating single object generation than multiple object generation. This may be because these metrics do not model the specific relationships between objects well. iii) For the sub-categories of single and multiple object generation, most metrics perform best on *Basic* and *Grouped*, respectively. It is reasonable because the evaluation needs to incorporate more factors when measuring complex scenes

Table S9. Ablation study for the input image types. Results of SRCC are reported.

SRCC	Alignment	Geometry	Texture	Overall
Texture	0.739	0.782	0.811	0.792
Normal	0.663	0.740	0.722	0.732
Texture + Normal	0.684	0.749	0.777	0.749

Table S10. Ablation study for the rotation angles. Results of SRCC are reported.

SRCC	Alignment	Geometry	Texture	Overall
-	0.739	0.782	0.811	0.792
30°	0.719	0.783	0.800	0.789
45°	0.723	0.789	0.802	0.793
60°	0.721	0.787	0.801	0.791

and relationships, which presents challenges for the existing metrics.

D.4. Performance Comparison on Different Generative Methods

In this section, we test the performance of different metrics on eight generative methods and report the results in Tab. S12. For the metrics that require fine-tuning, we perform a leave-one-method-out evaluation, that is, testing on one generative method while training on other seven methods, which can validate the generalization performance for unknown generative methods. From the table, we have the following observations: i) HyperScore achieves the best performance on the four dimensions of almost all generative methods, demonstrating its generalization capability to new generative methods. In contrast, other metrics show inconsistent results across different methods. ii) Almost all fine-tuned metrics exhibit a noticeable decline in performance when evaluating the two recent methods, 3DTopia and One-2-3-45++. This may be because the two methods present different distortion patterns compared to other methods. More specifically, 3DTopia tends to generate a single object, leading to comparatively lower scores in multiple object generation. One-2-3-45++ tends to generate flatter geometric structures and blurry textures. These deformations, which are not seen during training, are very challenging for evaluation. In contrast, fine-tuned metrics work well on DreamFusion because many other methods inherit DreamFusion’s paradigm and generate similar patterns. Therefore, it is important to consider the generalization towards new generative methods when designing text-to-3D evaluators.

References

- [1] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6
- [2] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4
- [3] Yuze He, Yushi Bai, Matthieu Lin, Wang Zhao, Yubin Hu, Jenny Sheng, Ran Yi, Juanzi Li, and Yong-Jin Liu. T³bench: Benchmarking current progress in text-to-3d generation. *arXiv preprint arXiv:2310.02977*, 2023. 3
- [4] Fangzhou Hong, Jiaxiang Tang, Ziang Cao, Min Shi, Tong Wu, Zhaoxi Chen, Tengfei Wang, Liang Pan, Dahua Lin, and Ziwei Liu. 3dtopia: Large text-to-3d generation model with hybrid diffusion priors. *arXiv preprint arXiv:2403.02234*, 2024. 2
- [5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [6] Three.js Library. <https://threejs.org>. 3
- [7] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2
- [8] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2
- [9] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *International Conference on Computer Vision (ICCV)*, 2023. 2
- [10] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2
- [11] OpenAI. Gpt-4 system card. *OpenAI*, 2023. 1
- [12] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations (ICLR)*, 2023. 2
- [13] ITU-R BT.500 Recommendation. Methodologies for the subjective assessment of the quality of television images. 2019. 3
- [14] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. In *International Conference on 3D Vision (3DV)*, 2024. 2
- [15] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Gregory Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2
- [16] Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v(ision) is a human-aligned evaluator for text-to-3d generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3
- [17] Zike Wu, Pan Zhou, Xuanyu Yi, Xiaoding Yuan, and Hanwang Zhang. Consistent3d: Towards consistent high-fidelity text-to-3d generation with deterministic sampling prior. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

Table S11. Performance comparison (in terms of SRCC) of different evaluators on eight prompt categories.

Metric	Alignment							
	Basic	Refined	Complex	Fantastic	Grouped	Action	Spatial	Imaginative
CLIPScore	0.574	0.592	0.407	0.421	0.492	0.434	0.404	0.591
BLIPScore	0.640	0.520	0.479	0.423	0.590	0.476	0.577	0.540
Aesthetic Score	0.221	0.285	0.283	0.051	0.053	0.086	0.013	0.062
ImageReward	0.775	0.672	0.544	0.528	0.631	0.622	0.626	0.608
DreamReward	0.671	0.597	0.427	0.424	0.483	0.454	0.448	0.491
HPS v2	0.597	0.348	0.146	0.388	0.562	0.250	0.407	0.518
CLIP-IQA	0.058	0.144	0.082	0.127	0.042	0.006	0.086	0.114
Q-Align	0.349	0.365	0.326	0.255	0.224	0.212	0.023	0.111
ResNet50 + FT	0.625	0.603	0.582	0.601	0.548	0.429	0.562	0.475
ViT-B + FT	0.552	0.533	0.622	0.519	0.609	0.623	0.656	0.483
SwinT-B + FT	0.626	0.541	0.651	0.536	0.455	0.508	0.570	0.556
DINO v2 + FT	0.669	0.649	0.720	0.665	0.643	0.651	0.594	0.598
MultiScore	0.680	0.693	0.710	0.621	0.655	0.615	0.625	0.580
HyperScore	0.819	0.690	0.733	0.705	0.701	0.681	0.637	0.691

Metric	Geometry							
	Basic	Refined	Complex	Fantastic	Grouped	Action	Spatial	Imaginative
CLIPScore	0.575	0.570	0.438	0.404	0.496	0.503	0.479	0.602
BLIPScore	0.667	0.491	0.496	0.398	0.556	0.510	0.611	0.574
Aesthetic Score	0.244	0.279	0.268	0.053	0.164	0.041	0.088	0.028
ImageReward	0.790	0.545	0.558	0.467	0.532	0.588	0.585	0.600
DreamReward	0.691	0.495	0.473	0.395	0.465	0.422	0.469	0.497
HPS v2	0.617	0.355	0.218	0.364	0.504	0.334	0.423	0.544
CLIP-IQA	0.107	0.215	0.178	0.133	0.092	0.107	0.199	0.029
Q-Align	0.423	0.469	0.439	0.334	0.444	0.396	0.111	0.079
ResNet50 + FT	0.693	0.671	0.712	0.655	0.641	0.491	0.638	0.515
ViT-B + FT	0.692	0.643	0.752	0.568	0.704	0.598	0.700	0.552
SwinT-B + FT	0.720	0.618	0.738	0.579	0.558	0.559	0.637	0.622
DINO v2 + FT	0.81	0.752	0.815	0.749	0.741	0.769	0.623	0.619
MultiScore	0.733	0.753	0.802	0.670	0.727	0.625	0.723	0.634
HyperScore	0.862	0.745	0.838	0.742	0.776	0.690	0.736	0.738

Metric	Texture							
	Basic	Refined	Complex	Fantastic	Grouped	Action	Spatial	Imaginative
CLIPScore	0.620	0.616	0.472	0.442	0.514	0.519	0.528	0.631
BLIPScore	0.707	0.556	0.586	0.458	0.543	0.507	0.665	0.584
Aesthetic Score	0.218	0.310	0.306	0.108	0.257	0.053	0.108	0.030
ImageReward	0.800	0.597	0.596	0.510	0.505	0.606	0.626	0.618
DreamReward	0.697	0.529	0.460	0.446	0.425	0.449	0.514	0.523
HPS v2	0.620	0.359	0.217	0.370	0.508	0.376	0.464	0.584
CLIP-IQA	0.128	0.228	0.230	0.215	0.111	0.165	0.223	0.023
Q-Align	0.469	0.532	0.485	0.425	0.524	0.474	0.231	0.211
ResNet50 + FT	0.721	0.703	0.733	0.658	0.654	0.686	0.553	0.477
ViT-B + FT	0.702	0.682	0.782	0.619	0.721	0.692	0.668	0.603
SwinT-B + FT	0.747	0.617	0.770	0.598	0.580	0.624	0.666	0.652
DINO v2 + FT	0.818	0.765	0.842	0.729	0.751	0.777	0.692	0.674
MultiScore	0.776	0.778	0.843	0.671	0.735	0.695	0.747	0.691
HyperScore	0.880	0.792	0.852	0.745	0.807	0.761	0.746	0.732

Metric	Overall							
	Basic	Refined	Complex	Fantastic	Grouped	Action	Spatial	Imaginative
CLIPScore	0.590	0.587	0.441	0.433	0.497	0.519	0.473	0.621
BLIPScore	0.678	0.508	0.502	0.433	0.570	0.507	0.626	0.579
Aesthetic Score	0.244	0.295	0.305	0.056	0.166	0.053	0.059	0.021
ImageReward	0.793	0.588	0.572	0.504	0.569	0.606	0.616	0.617
DreamReward	0.689	0.536	0.467	0.428	0.468	0.452	0.488	0.513
HPS v2	0.623	0.347	0.200	0.380	0.530	0.376	0.427	0.537
CLIP-IQA	0.096	0.202	0.163	0.163	0.064	0.165	0.155	0.023
Q-Align	0.424	0.465	0.439	0.345	0.406	0.349	0.131	0.09
ResNet50 + FT	0.690	0.675	0.695	0.640	0.611	0.608	0.508	0.512
ViT-B + FT	0.670	0.640	0.739	0.567	0.664	0.679	0.613	0.552
SwinT-B + FT	0.710	0.598	0.732	0.572	0.519	0.590	0.620	0.616
DINO v2 + FT	0.786	0.735	0.816	0.718	0.704	0.737	0.648	0.622
MultiScore	0.734	0.748	0.811	0.665	0.699	0.656	0.694	0.643
HyperScore	0.865	0.760	0.841	0.750	0.759	0.730	0.717	0.756

Table S12. Performance comparison (in terms of SRCC) of different evaluators on eight generative methods.

Metric	Alignment							
	DreamFusion	Magic3D	SJC	TextMesh	Consistent3D	LatentNeRF	3DTopia	One-2-3-45++
CLIPScore	0.500	0.337	0.514	0.356	0.242	0.431	0.411	0.361
BLIPScore	0.550	0.390	0.520	0.443	0.337	0.527	0.502	0.337
Aesthetic Score	0.076	0.092	0.088	0.074	0.057	0.029	0.361	0.074
ImageReward	0.677	0.538	0.585	0.610	0.508	0.632	0.613	0.547
DreamReward	0.534	0.396	0.333	0.545	0.283	0.324	0.577	0.464
HPS v2	0.497	0.275	0.316	0.420	0.249	0.295	0.340	0.267
CLIP-IQA	0.223	0.163	0.040	0.179	0.023	0.122	0.094	0.137
Q-Align	0.079	0.093	0.096	0.093	0.021	0.231	0.076	0.13
ResNet50 + FT	0.689	0.508	0.450	0.530	0.430	0.401	0.259	0.222
ViT-B + FT	0.726	0.582	0.475	0.652	0.389	0.396	0.376	0.155
SwinT-B + FT	0.722	0.560	0.347	0.649	0.442	0.462	0.148	0.223
DINO v2 + FT	0.791	0.633	0.609	0.72	0.501	0.472	0.45	0.259
MultiScore	0.784	0.670	0.657	0.597	0.672	0.493	0.365	0.415
HyperScore	0.846	0.754	0.724	0.816	0.724	0.697	0.599	0.573

Metric	Geometry							
	DreamFusion	Magic3D	SJC	TextMesh	Consistent3D	LatentNeRF	3DTopia	One-2-3-45++
CLIPScore	0.511	0.323	0.493	0.343	0.226	0.418	0.360	0.269
BLIPScore	0.549	0.356	0.537	0.431	0.328	0.553	0.426	0.316
Aesthetic Score	0.159	0.156	0.088	0.051	0.015	0.005	0.452	0.367
ImageReward	0.665	0.461	0.588	0.552	0.433	0.555	0.503	0.421
DreamReward	0.513	0.369	0.363	0.533	0.237	0.333	0.502	0.394
HPS v2	0.506	0.299	0.311	0.429	0.252	0.346	0.309	0.331
CLIP-IQA	0.202	0.099	0.014	0.142	0.026	0.085	0.099	0.078
Q-Align	0.022	0.048	0.118	0.019	0.01	0.321	0.073	0.449
ResNet50 + FT	0.707	0.537	0.482	0.545	0.443	0.439	0.429	0.358
ViT-B + FT	0.743	0.563	0.582	0.629	0.375	0.466	0.546	0.215
SwinT-B + FT	0.729	0.529	0.382	0.602	0.434	0.525	0.369	0.184
DINO v2 + FT	0.775	0.624	0.681	0.73	0.554	0.603	0.69	0.471
MultiScore	0.741	0.620	0.595	0.585	0.581	0.446	0.588	0.418
HyperScore	0.821	0.717	0.703	0.773	0.657	0.708	0.683	0.653

Metric	Texture							
	DreamFusion	Magic3D	SJC	TextMesh	Consistent3D	LatentNeRF	3DTopia	One-2-3-45++
CLIPScore	0.502	0.406	0.509	0.393	0.259	0.434	0.384	0.258
BLIPScore	0.542	0.400	0.532	0.462	0.330	0.536	0.473	0.298
Aesthetic Score	0.141	0.122	0.095	0.020	0.003	0.047	0.450	0.356
ImageReward	0.671	0.489	0.573	0.568	0.435	0.535	0.543	0.420
DreamReward	0.498	0.377	0.364	0.517	0.251	0.281	0.533	0.364
HPS v2	0.516	0.295	0.330	0.450	0.266	0.307	0.367	0.307
CLIP-IQA	0.195	0.101	0.024	0.119	0.006	0.169	0.093	0.007
Q-Align	0.018	0.072	0.19	0.021	0.024	0.388	0.187	0.44
ResNet50 + FT	0.702	0.532	0.450	0.553	0.421	0.432	0.395	0.412
ViT-B + FT	0.735	0.606	0.572	0.636	0.392	0.483	0.492	0.220
SwinT-B + FT	0.734	0.524	0.374	0.617	0.459	0.519	0.340	0.171
DINO v2 + FT	0.792	0.635	0.659	0.736	0.563	0.581	0.663	0.423
MultiScore	0.755	0.634	0.603	0.598	0.593	0.472	0.610	0.448
HyperScore	0.823	0.736	0.694	0.775	0.702	0.694	0.613	0.623

Metric	Overall							
	DreamFusion	Magic3D	SJC	TextMesh	Consistent3D	LatentNeRF	3DTopia	One-2-3-45++
CLIPScore	0.506	0.342	0.517	0.362	0.208	0.428	0.399	0.330
BLIPScore	0.548	0.380	0.553	0.443	0.300	0.543	0.473	0.347
Aesthetic Score	0.118	0.149	0.066	0.005	0.001	0.029	0.434	0.242
ImageReward	0.678	0.504	0.602	0.577	0.440	0.572	0.555	0.517
DreamReward	0.522	0.391	0.356	0.548	0.237	0.332	0.544	0.462
HPS v2	0.507	0.291	0.337	0.436	0.241	0.318	0.336	0.322
CLIP-IQA	0.206	0.135	0.023	0.156	0.005	0.111	0.104	0.028
Q-Align	0.038	0.013	0.154	0.024	0.048	0.297	0.124	0.327
ResNet50 + FT	0.703	0.543	0.470	0.535	0.525	0.432	0.357	0.302
ViT-B + FT	0.731	0.597	0.561	0.635	0.388	0.439	0.460	0.190
SwinT-B + FT	0.733	0.567	0.363	0.639	0.441	0.499	0.297	0.230
DINO v2 + FT	0.791	0.65	0.686	0.724	0.543	0.567	0.61	0.363
MultiScore	0.769	0.659	0.635	0.597	0.613	0.451	0.508	0.444
HyperScore	0.839	0.745	0.730	0.794	0.698	0.718	0.638	0.634



Please assist us in creating a set of text prompts for text-to-3D generation. To ensure diversity and comprehensiveness in the prompts, the content should focus on the following four aspects. Here, we provide the definitions below:

1. 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 properties:

- 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.
- Size: Small, medium, large, specific dimensions

3. Appearance properties:

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

4. Object interaction:

- Spatial relation: the orientation or movement involved in the interaction, describing where the subject and object are in relation to each other. e.g. toward, above, below, near to, on the left/right/top/bottom of
- Action relation: Specific verbs representing interactions between subjects and objects, e.g. wear, watch, speak to, hold, run, jump, kiss, sit on, stand on, walk with, play with

Then, we will give you some more tips and requirements for the prompt creation. Here, we will present different categories for single object generation and multiple object generation, respectively.

For single object generation, we define four categories:

- 1. Basic:** descriptions about a single object without detailed geometry or appearance details, where some global properties (e.g., color, shape, material) could be included. Here are examples: "A green apple", "A humanoid robot", "A square table", "A wooden chair".
- 2. Refined:** descriptions about a single object with one simple specification of the geometry or appearance details. Here are examples: "A photo of a black bird", "A yellow rubber duck has only one foot", "A plastic cup with a printed logo", "An imperial state crown of England".
- 3. Complex:** descriptions about a single object with more than two detailed specifications of geometry or appearance details. Here are examples: "A brown teddy bear, fur matted, one eye missing", "A turtle standing on its hind legs, wearing a top hat and holding a cane", "An oversized, porous, sphere-shaped birdcage, made of woven golden wires".
- 4. Fantastical:** descriptions about a single object with high creativities that are not, or are generally unlikely to be, found in the real world. Here are examples: "A frog with a translucent skin displaying a mechanical heart beating", "A squirrel-octopus hybrid", "A white cat has three tails, made of white, purple and black crystals", "A tiger dressed as a doctor".

For multiple object generation, we define four categories:

- 1. Grouped:** descriptions about multiple objects without interactions, where some global properties (e.g., color, shape, material) could be included. Here are examples: "A delicious hamburger and a green apple", "A red pig and a huge drum", "A round table and a square chair".
- 2. Action:** descriptions about multiple objects with interaction about action relations, where some geometry or appearance details could be included. Here are examples: "A humanoid robot with a top hat is playing the cello", "A black cat with white feet is sleeping peacefully beside a carved pumpkin", "A dog is eating a red apple with its tail raised".
- 3. Spatial:** descriptions about multiple objects with interactions about spatial relations, where some geometry or appearance details could be included. Here are examples: "A red apple on a round ceramic plate", "A strong football player wearing a number seven jersey next to a blue soccer", "A blue rose in a crystal, symmetrical vase".
- 4. Imaginative:** descriptions about multiple objects with interactions, where objects or interactions are not, or are generally unlikely to be found in the real world. Here are examples: "A panda with a wizard hat is reading a newspaper", "A ghost is eating a golden apple", "A lion wearing a crown is painting with a glowing paintbrush", "A raccoon astronaut is holding his helmet".

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



Got it! You've outlined a detailed framework with distinct categories, geometry and appearance properties, and types of interactions for both single and multiple object generation. Here's a quick summary to confirm my understanding of what you need:

Object Categories: Diverse sets of real and abstract objects—ranging from living beings to food.

Geometry Properties: Details like symmetry, contours, shape, and volume to define the structural features.

Appearance Properties: Including colors, materials, textures, and finish to dictate the look and feel of objects.

Object Interaction: Focus on spatial and action-related interactions between objects, especially for multiple object generation.

Then, the challenges break down as follows:

Single Object Generation:

Basic: Simple object descriptions with minimal detail.

Refined: One additional property for a bit more complexity.

Complex: Multiple geometry or appearance properties for detailed descriptions.

Fantastical: Creative, unreal objects that may blend features or stretch reality.

Multiple Object Generation:

Grouped: Multiple objects without interaction.

Action: Interactions involving actions, like movement or usage.

Spatial: Objects with spatial relations to each other.

Imaginative: Unreal interactions or fantastical scenes.

This structure ensures a diverse and comprehensive prompt set, with various levels of detail and creativity across single and multiple object descriptions. I'm ready for the next steps, such as requirements for the prompt creation. Please provide the additional information or tips you mentioned.

Figure S8. Examples of guiding GPT-4 for automatic prompt generation. (Part 1/2)

Please create 100 "Basic" prompts about "single object generation" within 10 words. Do not describe the environment. Please make sure that the prompt you provided has enough diversity and comprehensiveness, and that the format is similar to my example.



Here are 100 "Basic" prompts about a single object, each within 10 words, ensuring a wide range of categories and diversity:

1. A striped zebra
2. A green pepper
3. A brown leather belt
4. A glowing light bulb
5. A blue vase
6. A round stool
7. A wooden boat
8. A white wedding dress
9. A round gold coin
10. A silver spoon
11. A wooden dining table
12. A yellow pencil
13. A purple notebook
14. ...
100. A delicious hamburger

These prompts span a variety of categories, including clothing, accessories, household items, vehicles, and natural elements, ensuring coverage of different forms, materials, and textures.

Please create 100 "Spatial" prompts about "multiple object generation" within 20 words. Do not describe the environment. Please make sure that the prompt you provided has enough diversity and comprehensiveness, and that the format is similar to my example.



Here are 100 "Spatial" prompts about multiple objects, each within 20 words, focusing on spatial relationships between objects with diversity and clarity:

1. A red apple on a white ceramic plate
2. A black hat hanging on a brass hook
3. A glass teapot next to a porcelain cup
4. A golden key inside a brown leather pouch
5. A blue backpack under a wooden bench
6. A silver spoon resting on a folded napkin
7. A cat sitting next to a pair of shoes
8. A guitar leaning against a wooden chair
9. A clock above a wooden bookshelf
10. A pair of sunglasses on top of a magazine
11. A candle placed inside a glass jar
12. A handwoven basket filled with a ripe peach
13. A set of dice beside a deck of cards
14. ...
100. A small cactus inside a clay pot

These prompts ensure spatial relationships between objects are clear, diverse, and easy to visualize while maintaining a compact, simple structure.

Please create 100 "Imaginative" prompts about "multiple object generation" within 20 words. Do not describe the environment. Please make sure that the prompt you provided has enough diversity and comprehensiveness, and that the format is similar to my example.



Here are 100 "Imaginative" prompts about multiple objects, each within 20 words, ensuring diversity, creativity, and intriguing interactions between objects:

1. A giraffe wearing sunglasses plays chess with a robot
2. A dragon sipping tea from a porcelain cup
3. A violin floating next to a glowing crystal ball
4. A penguin in a tuxedo is conducting an orchestra with a baton
5. A ghost riding a bicycle with a pumpkin helmet
6. A knight polishing his sword while a cat reads a map
7. A frog wearing boots and holding an umbrella
8. A unicorn carrying a glowing lantern on its horn
9. A lion rider is riding a motorcycle
10. A shark with a crown is holding a golden trident
11. A bear astronaut is planting a flag with a paw print
12. A phoenix with flaming wings is carrying a silver lantern
13. A raccoon detective is studying a map under a magnifying glass
14. ...
100. A dog wearing backpack is walking on a rainbow

These imaginative prompts feature whimsical, surreal interactions between objects and creatures, maintaining variety and unexpected combinations for creativity in 3D generation.

Figure S9. Examples of guiding GPT-4 for automatic prompt generation. (Part 2/2)


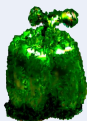



















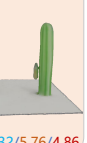





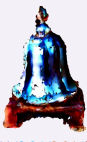









	DreamFusion	Magic3D	SJC	TextMesh	3DTopia	Consistent3D	LatentNeRF	One2-3-45++
Basic	A green pepper							
								
	8.74/8.39/7.39/8.00	8.89/8.73/7.67/8.43	5.00/3.28/2.80/3.71	9.53/8.95/8.44/8.90	6.41/5.42/5.81/5.74	5.74/5.00/4.72/4.90	3.56/2.10/2.82/2.58	8.85/8.76/7.88/8.47
Refined	A cactus with a blooming flower							
								
	6.58/5.45/5.57/5.76	7.58/5.63/5.72/6.10	6.82/4.89/4.44/5.32	6.17/5.10/5.12/5.53	6.41/5.42/5.81/5.74	3.16/2.67/2.50/2.62	7.00/6.41/5.53/6.11	6.89/4.94/4.75/5.33
Complex	A blue jacket with silver buttons, fur lining on the collar, and a tear on the right sleeve							
								
	4.89/4.17/4.28/4.29	7.00/6.89/6.75/7.05	1.06/1.19/1.29/1.11	7.26/6.75/6.78/7.00	6.11/6.00/5.55/5.76	5.56/4.90/4.94/4.95	2.63/2.06/2.05/1.95	6.79/7.33/7.00/7.10
Fantastical	A house shaped like a giant teapot, complete with handle and spout							
								
	4.89/5.06/4.70/4.38	4.42/4.50/3.44/3.76	4.89/3.28/3.39/3.48	4.89/5.47/3.94/4.29	6.33/6.06/5.10/5.68	5.11/5.47/4.61/4.57	3.76/2.58/2.94/2.58	6.35/6.42/6.38/6.00
Group	A green cactus and a glass bottle							
								
	5.12/6.00/6.13/5.47	5.90/7.88/7.59/6.95	5.89/5.63/4.85/5.20	5.42/7.00/6.80/5.90	4.24/4.19/4.45/4.21	4.90/4.41/4.29/4.16	3.53/2.68/3.06/2.84	4.32/5.32/5.76/4.86
Action	A duck is chasing a yellow ball with its wings flapping							
								
	5.94/5.47/5.50/5.11	6.05/4.06/3.89/4.33	5.16/3.73/3.24/3.86	4.21/3.50/3.78/3.52	1.58/1.58/1.78/1.38	3.89/3.58/3.67/3.48	6.86/4.75/4.90/5.53	7.61/6.47/6.25/6.55
Spatial	A polished silver bell resting on a wooden stand							
								
	6.68/6.50/5.61/6.24	7.00/6.78/6.39/6.48	4.41/2.84/2.81/3.16	7.11/7.21/6.89/7.00	4.38/4.95/5.50/4.50	5.81/5.44/4.65/5.26	5.06/3.47/3.56/3.79	7.63/8.00/8.00/7.86
Imaginative	A mermaid is holding a white conch shell							
								
	7.47/6.05/5.33/6.10	7.11/5.89/5.90/6.48	5.79/3.55/2.95/3.67	5.53/4.17/3.95/4.52	2.79/2.89/3.05/2.60	2.47/1.91/2.43/1.76	6.95/5.83/4.74/6.05	7.63/6.22/5.39/6.29

Figure S10. More results of eight generative methods for eight categories. The scores below each sample denote alignment, geometry, texture, and overall quality, respectively.