Think-Then-Generate: Structural Chain-of-Thought Reasoning for Consistent 3D Generation

Supplementary Material

In this document, we provide a more detailed description of the model and implementation details to complement the main content. Additionally, we include extra experiments and prompt templates.

6. Implementation Details

6.1. Inference Pipeline

Algorithm 1: Thoughtful3D Generation Framework

Require: Text prompt P
Ensure: 3D model $\mathcal{M}$

// Before Generation: Planning Phase
semantics $\leftarrow$ SemanticParser(P) {Extract objects/attributes}
timeline $\leftarrow$ Planner$(P, \text{semantics})$ {Plan generation stages}
// During Generation: Optimization Phase
Initialize 3D model $\mathcal{M}$
for optimization step $s \in$ timeline **do**
  Strategy 1: 3DRefine-CoT
  Render views $\mathcal{V} = \{v_i\}$ from $\mathcal{M}$
  $\mathcal{L}_{\text{recon}} \leftarrow 0$
  for each view $v_i \in \mathcal{V}$ **do**
    $(\text{score}_i, \text{caption}_i) \leftarrow \text{Verifier}(v_i, P)$
    if $\text{score}_i <$ threshold **then**
      Generate corrections $\mathcal{C}_i$ using diffusion model
      $v_i' \leftarrow \text{ConsensusSelector}(\mathcal{C}_i)$
      $\mathcal{L}_{\text{recon}} \leftarrow \mathcal{L}_{\text{recon}} + \text{MSE}(v_i, v_i')$
    end if
  end for
  Strategy 2: Cross-View Alignment
  $\mathcal{L}_{\text{align}} \leftarrow \text{ComputeAlignmentLoss}(\mathcal{V})$
  Joint Optimization
  $\mathcal{L}_{\text{SDS}} \leftarrow \text{ScoreDistillationLoss}(\mathcal{V})$
  $\mathcal{L}_{\text{total}} \leftarrow \lambda_1 \mathcal{L}_{\text{SDS}} + \lambda_2 \mathcal{L}_{\text{align}} + \lambda_3 \mathcal{L}_{\text{rc}}$
  Update ϕ via $\nabla_\phi \mathcal{L}_{\text{total}}$
end for
return $\mathcal{M}$

6.2. Camera pose setting

In 3DRefine-CoT, to ensure that the rendered images $\{V_i\}_{i=1}^N$ preserve both distinguishable global structures and local details, we introduce an additional set of N camera viewpoints $\{C_i\}_{i=1}^N$ (with $N = 4$ in our experiments) arranged in a spherical coordinate system. The rendered views generated from these additional cameras are used in the subsequent CoT reasoning process. They serve as crucial evidence for identifying inconsistencies or other issues that may arise during the 3D generation process.

6.3. Hyperparameter Settings

In our experiment, based on the magnitude of each loss term, we set the coefficients $\{\lambda_1, \lambda_2, \lambda_3\}$ defined in Eq.(13) as $\lambda_1 = 1$ for $\mathcal{L}_{SDS}$, $\lambda_1 = 0.3$ for $\mathcal{L}_{rc}$, $\lambda_3 = 1$ for $\mathcal{L}_{align}$. We incorporate the $\mathcal{L}_{rc}$ and $\mathcal{L}_{align}$ losses only after the 3D model has developed an initial structural form, typically during the mid-to-late stages defined by 3DBlueprint-CoT.

7. Additional Experiments

7.1. Comparison with other 3D Consistency Enhancement methods

We demonstrate the superior performance of our Thoughtful3D compared to other approaches aimed at improving 3D consistency. Here, we present quantitative and qualitative comparisons with Perp-Neg, Debias, Hallo3D. As shown in Fig. 8, Thoughtful3D not only effectively addresses the Janus problem, but also significantly improves the quality of generated objects. Specifically, compared to prior methods, Thoughtful3D generates pandas with multi-view consistency and without Janus problem. For storks, it produces structurally complete and realistic shapes, while other methods exhibit varying degrees of structural abnormalities in these cases. A substantial improvement is also observed in the quantitative metric, CLIP-Score.

7.2. A Case Study Assessing CoT Reasoning Capabilities in 3D Generation Tasks

Here, by comparing direct LLM image queries with CoT reasoning analysis, we demonstrate that CoT reasoning can effectively locate "hallucination" phenomena in renderings, such as structural errors, repetitive features, and misalignments. Using the prompt "An elegant flamingo standing tall with long legs and pinkish-white feathers." as an example, we selected one rendered view from its training process for analysis. As shown in Fig. 9, when directly queried with an LLM, the model can identify some existing issues. However, it fails to precisely detect a key problem: an abnormally distorted leg structure. In contrast, our structural CoT reasoning accurately identifies this critical issue, as demonstrated in Fig. 10.

In structural CoT reasoning, image analysis is carried out in two stages. In the first stage, the image is carefully

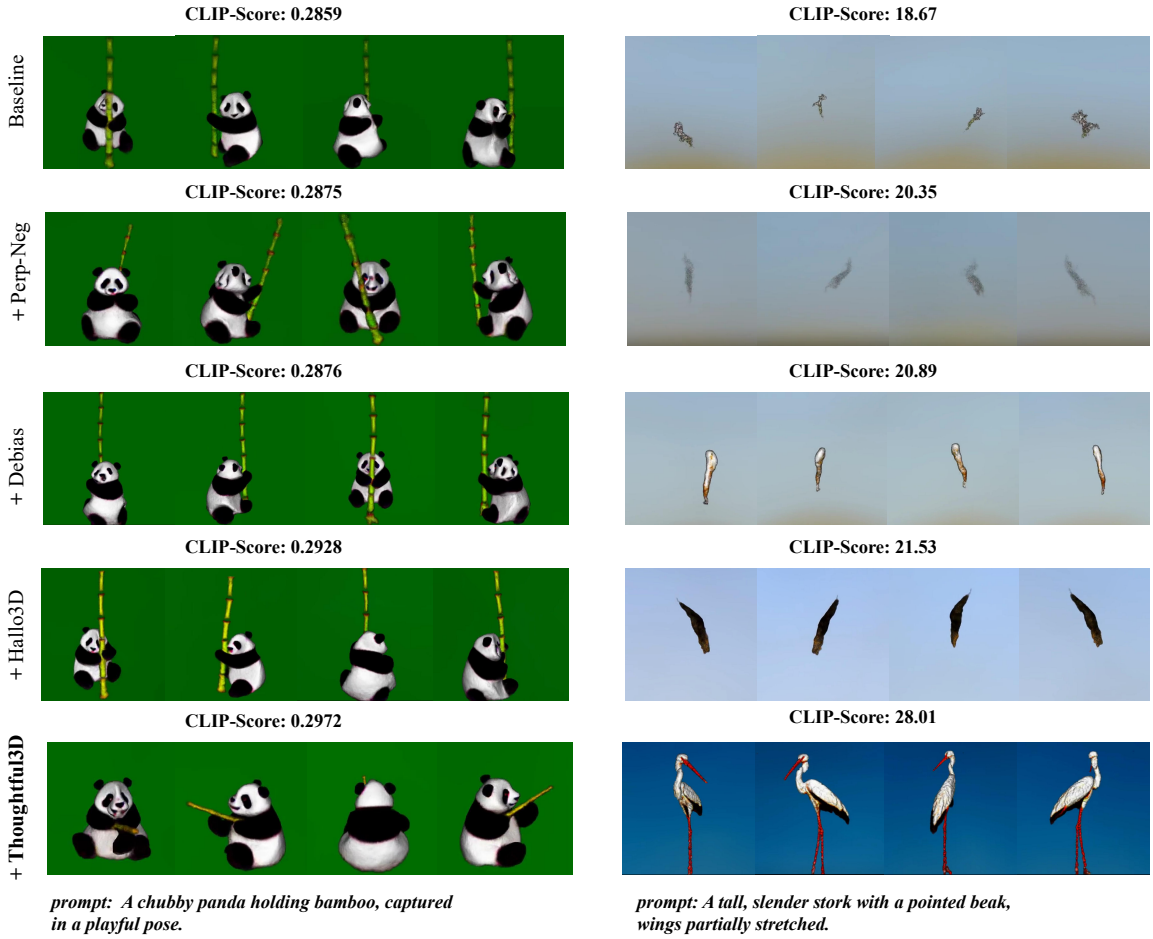

Figure 8. Comparison experiments with Perp-Neg, Debias and Hallo3D.

described and scored. In the second stage, both the image and its description are used together to analyze the problem from different perspectives. The main issue is then identified. By adopting a "description then decision" strategy, perception is separated from reasoning during analysis. This improves interpretability, matches human cognitive processes, and greatly enhances the ability to analyze rendered images in 3D generation.

7.3. Additional Qualitative Results

Fig. 11 and Fig. 12 present qualitative results of our method integrated with different baseline models. In each image pair, the left side shows renderings of 3D models generated by the original approach, while the right side displays results from our method. Through planning, reflection, and correction during 3D model generation, Thoughtful3D significantly improves multi-view coherence and enhances output quality. Consequently, it achieves 3D generation that faithfully aligns with diverse input prompts.

7.4. Robustness Analysis

We address MLLM stochasticity by employing Consensus Selector as a *robust filter*. *1) High Reliability.* A user study (300 pairs) shows 87% feedback accuracy and only 5% hallucination (Tab. 4). *2) Error Filtering.* To test sensitivity to errors, we injected 20% random/incorrect negative prompts. The Consensus Selector effectively *intercepted these errors*, raising the discard rate to 28% (Tab. 5) and preventing model degradation. *3) Stability.* By aggregating independent MLLM votes, our method reduces cross-run variance by 44% compared to a single-model setup (Tab. 6), ensuring stable 3D generation despite MLLM stochasticity.

Table 4. User Study of MLLM feedback.

Neg. Prompt	Count	Percentage
Accurate	261	87%
Subjective	24	8%
Hallucination	15	5%

Table 5. MLLM Consensus Selector discard rate.

Neg. Prompt	CLIP	Discard Rate
Standard (Ours)	**30.11**	**10%**
Noise (Random)	29.45	21%
Adversarial (Wrong)	29.02	28%

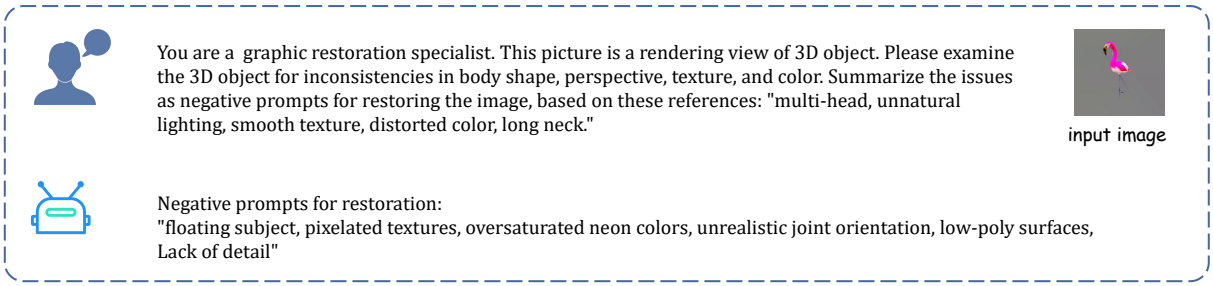

Figure 9. Analyze rendered images directly using MLLMs.

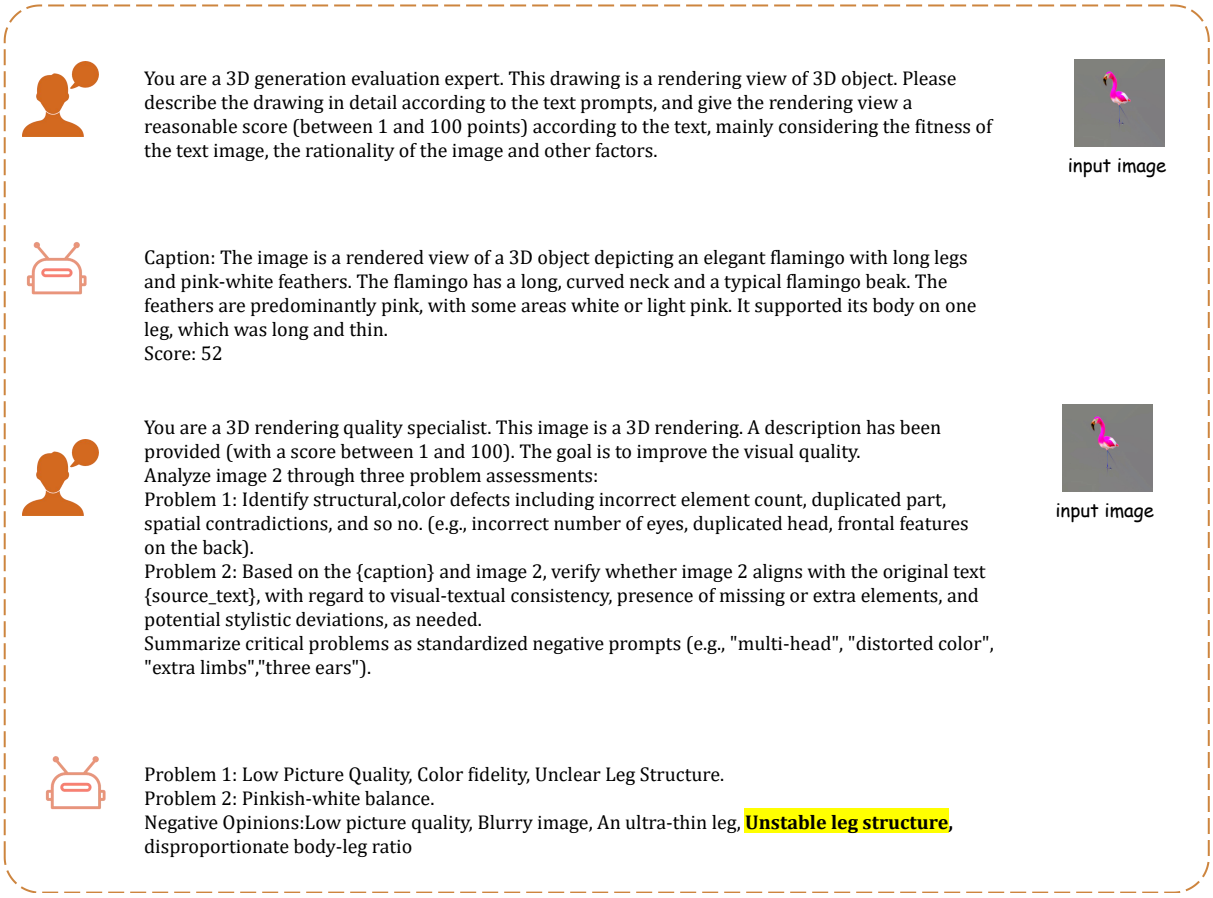

Figure 10. Analyze rendered images through structural CoT Reasoning.

Table 6. Consistency across multiple generations.

Method	CLIP ↑	Avg. Std ↓	Std@95% ↓	Run-to-Run Sim ↑
w/o consensus (Single)	0.283 ± 0.018	0.018	0.027	93.75
Ours (K=3 + Consensus)	**0.302 ± 0.010**	**0.010**	**0.018**	**96.48**

7.5. Efficiency Comparison

As shown in Tab. 7, we report the runtime on an NVIDIA A100 GPU using *GaussianDreamer* and *DreamFusion* as baselines. Although our method adds computational overhead, we consider it justified by the quality gains. We ensure efficiency by applying 3DRefine-CoT exclusively in later stages to refine established geometry, striking an optimal balance.

Table 7. Analysis of runtime and memory usage.

Model	CLIP (Baseline)	CLIP (Ours)	Iteration	Refine Start	Org. Time	Extra Time
GaussianDreamer	27.33	**30.39**	1200	900	∼15 min	∼5 min
DreamFusion	23.22	**28.67**	2500	2100	∼27 min	∼6 min

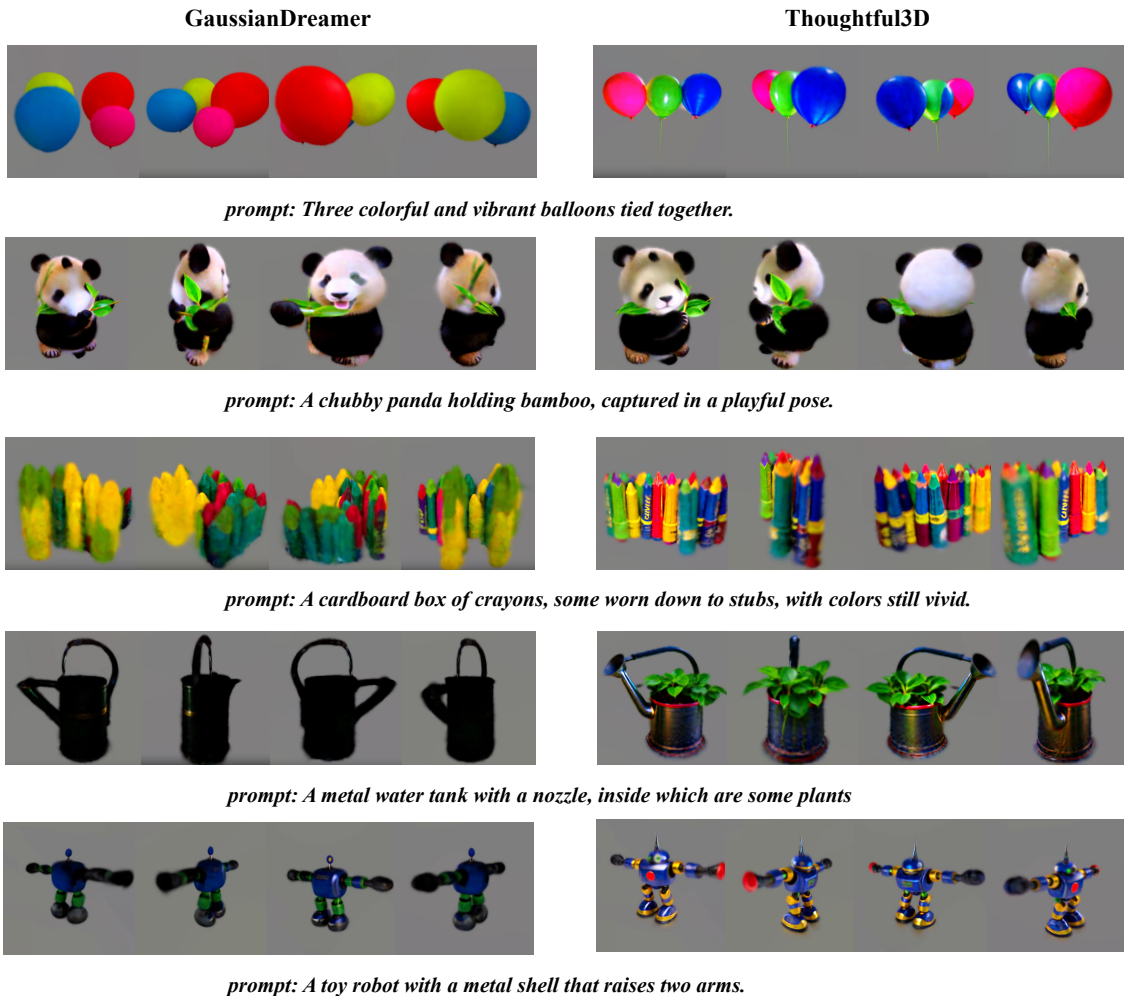

Figure 11. Analyze rendered images directly using MLLMs.

7.6. MLLM Sensitivity

As shown in Tab. 8, we evaluate our method across multiple open- and closed-source MLLM. performance differences among other high-capacity models like GPT-5.1 and Qwen2.5-VL-72B are minimal, suggesting low model sensitivity. Overall, our method remains stable across strong MLLMs and benefits from more capable models.

Table 8. Performance comparison across MLLMs.

Model	GaussianDreamer			DreamFusion		
	B/32	B/16	L/14	B/32	B/16	L/14
Baseline	21.45	26.71	27.33	16.40	22.98	23.22
GPT-4o (Ours)	**24.65**	**29.15**	**30.39**	**21.98**	**27.87**	**28.67**
GPT-5.1	25.17	30.21	30.87	22.36	29.01	29.94
Qwen2.5-VL-72B-Instruct	22.87	28.79	29.42	21.44	27.13	28.52

Table 9. Ablation on the number of candidate negative prompts.

	K=1	K=2	K=3	K=4	K=6	K=8
CLIP	28.43	29.62	**30.15**	30.21	30.17	30.20
Time (min)	16.7	18.3	**20.4**	23.5	30.1	36.9
ΔCLIP/ΔTime	-	0.744	**0.252**	0.019	0.006	0.004

7.7. Discussion on the test-time scaling

As shown in Tab. 9, we study the effect of the number of candidate negative prompts in 3DRefine-CoT. The metric $\Delta\text{CLIP}/\Delta\text{Time}$ measures the marginal CLIP gain per additional minute relative to the previous K. $K = 3$ yields the highest gain per minute, while further increasing K (e.g., $K = 6$) incurs higher computational cost but leads to degraded performance, indicating that additional candidates may introduce noise or negative effects. We therefore set $K = 3$ as the best efficiency–performance trade-off.

7.8. Transition window for prompt switching.

As mentioned earlier, the transition window during prompt switching is utilized in our 3DBlueprint-CoT process. The

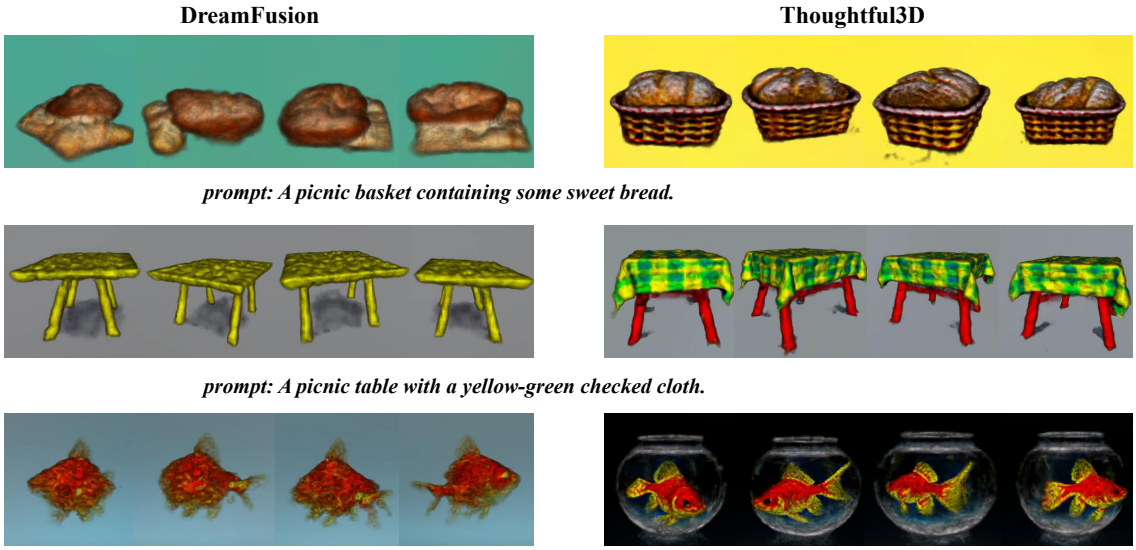

Figure 12. Analyze rendered images directly using MLLMs.

switching procedure for dynamic prompts can be expressed as:

$$p_k(s) = \begin{cases} S_o, & s \in \tau_1, \\ S_c + \lambda_a(s) S_a, & s \in \tau_2, \\ S_c + S_a + \lambda_h(s) S_h, & s \in \tau_3, \end{cases} \tag{14}$$

The weighting $\lambda_a(s)$ and $\lambda_h(s)$ coefficients are defined as:

$$\lambda_a(s) = \begin{cases} 0, & s \le T_1, \\ \frac{s - T_1}{\Delta T}, & T_1 < s \le T_1 + \Delta T, \\ 1, & s > T_1 + \Delta T, \end{cases} \tag{15}$$

$$\lambda_h(s) = \begin{cases} 0, & s \le T_2, \\ \frac{s - T_2}{\Delta T}, & T_2 < s \le T_2 + \Delta T, \\ 1, & s > T_2 + \Delta T. \end{cases} \tag{16}$$

where ΔT represents the width of the transition window during prompt switching, defined as the number of training iterations required for smooth transitions between stages. In our experiments, this parameter is typically set to 100.

8. Prompt Templates

We present the core prompts used in this work, specifically those applied in 3DBlueprint-CoT and 3DRefine-CoT, respectively. Fig. 13 and Fig. 15 belong to the 3DBlueprint-CoT. Fig. 13 parses the input prompt, while Fig. 15 plans the specific generation process. Fig. 16 and Fig. 14 belong to the 3DRefine-CoT. Fig. 16 provides a detailed description and scoring of the rendered image. Fig. 14 analyzes issues and summarizes findings for the rendered image.

You are a prompt semantic expert.
Analyze the 3D generation prompt through systematic linguistic decomposition.
The input query is: "{query}". Please analyze which words belong to the main structure of the sentence, which words are adjectives, and which words are abstract terms.

Reference Analysis:
Here is a reference example: For the sentence "An elegant flamingo standing tall with long legs and pinkish-white feathers." The main structure is: "A flamingo standing with legs and feathers.". The adjectives are:"tall, long, pinkish-white". The abstract word is: "elegant".
then the returned result is:
Structure: "A flamingo standing with legs and feathers."
Adjectives: "tall, long, pinkish-white"
Abstract: "elegant"

Output Format:
Structure: "[Core structural sentence]"
Adjectives: "[Comma-separated modifiers]"
Abstract: "[Comma-separated abstract terms]"
Please think carefully and provide your response.

Figure 13. Prompt of Semantic Parsing

9. limitations

Our approach functions as an optimization strategy built upon a baseline model, wherein the final output quality is directly contingent upon the baseline's performance. Critically, when the baseline exhibits suboptimal performance, our method demonstrates limited efficacy in substantially enhancing the quality and consistency of the generated 3D

You are a 3D rendering quality specialist. Two images are provided: image 1 is a composite of high-quality renderings of "{source_text}", and image 2 is the target view described as "{caption}" with a current score of {score}/100 requiring improvement. Analyze image 2 through three problem assessments:
Problem 1: Identify structural,color defects including incorrect element count, duplicated part, spatial contradictions, and so no. (e.g., incorrect number of eyes, duplicated head, frontal features on the back).
Problem 2: Based on the {caption} and image 2, verify whether image 2 aligns with the original text {source_text}, with regard to visual-textual consistency, presence of missing or extra elements, and potential stylistic deviations, as needed.
Problem 3: Compare image 2 with the reference rendering in image 1 to identify structural inconsistencies or abnormalities, such as additional ears, distorted limbs, or duplicated facial features.
Summarize critical problems as standardized negative prompts (e.g., "multi-head", "distorted color", "extra limbs","three ears").

Output Format:
Problem 1: [findings], Problem 2: [findings], Problem 3: [findings], Negative Prompts: [comma-separated descriptors]
Please think carefully and provide your response.

Figure 14. Prompt of Summarize the negative prompts

content.

You are a linguistic reconstruction expert. Reconstruct the input prompt through three progressive enrichment stages using the provided structural analysis.
Input Components: Original prompt "{query}" and Structural analysis "{structure}".
Reconstruction Protocol:
Stage 1 (Core Skeleton): Extract essential syntactic elements (primary subject nouns, key action verbs, critical spatial prepositions) to form a minimal grammatical sentence.
Stage 2 (Concrete Enhancement): Integrate concrete descriptors (physical attributes, quantitative specifications, material indicators) while maintaining syntax.
Stage 3 (Abstract Enrichment): Incorporate non-tangible elements (subjective qualities, stylistic indicators, atmospheric cues) to achieve near-perfect semantic equivalence.
Reference Implementation:
Input Query: "An elegant flamingo standing tall with long legs and pinkish-white feathers."
Structural Analysis:
Structure: "A flamingo standing with legs and feathers"
Adjectives: tall, long, pinkish-white
Abstract: elegant
Output:
Stage 1: A flamingo standing with legs and feathers.
Stage 2: A flamingo standing with long legs and pinkish-white feathers.
Stage 3: An elegant flamingo standing tall with long legs and pinkish-white feathers.

Output Format Requirements:
Strictly use this sequence format:
Stage 1: [Core skeleton reconstruction]
Stage 2: [Concrete-enhanced reconstruction]
Stage 3: [Abstract-enriched reconstruction]
Please think carefully and provide your response.

Figure 15. Prompt of Generation Planning

You are a 3D rendering evaluation expert.
Analyze the image of the 3D object "{text}" by providing a comprehensive description covering geometric structure, proportions, material properties, surface textures, camera perspective, spatial relationships, and fidelity to textual specifications.
Then perform quantitative evaluation (1-100 scale) using weighted criteria: Text-Image Alignment (50% weight, evaluating accuracy of entities and attributes, spatial consistency, and style adherence) and Visual Plausibility (50% weight, assessing structural coherence, absence of artifacts, and perspective accuracy).
Calculate the final score as (Alignment Score × 0.5) + (Plausibility Score × 0.5).

Output the results as: Description: [your analysis] Keep the description within 100 words.
Score: [final score to one decimal place].
Please think carefully and provide your response.

Figure 16. Prompt of Reflection