STRUCT2REAL: A SYSTEMATIC FRAMEWORK FOR ACCURATE AND EFFICIENT STRUCTURE-GROUNDED OBJECT IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in image generation have enabled the creation of high-quality visual content with impressive semantic fidelity. However, generating object images under fine-grained structural constraints, particularly preserving topology and spatial layout, remains an open challenge. We propose **Struct2Real**, a structure-grounded generation framework that enables photorealistic object image synthesis under explicit structural control while supporting flexible viewpoint and structural editing, consisting of twofold. **1)** we develop a **structure modeling system** that enables users to create a 3D structural representation named StructMap, which is composed of various geometric primitives and allows precise, intuitive and editable structural specification. **2)** We develop the **Structure-Aware Generation Agent (SAGA)**, a skill-oriented agent that formulates structure-grounded image synthesis as a dynamic skill orchestration process and enables photorealistic object image generation under structural constraints encoded in StructMap. SAGA further incorporates a realism memory module that retains visual attributes throughout generation, enabling viewpoint and structural editing with consistent visual attributes. Extensive experiments demonstrate that Struct2Real achieves strong performance in structure-grounded object image generation while ensuring low user effort required for this task, highlighting the practicality and effectiveness of our method. *Please refer to more details in the Appendix and supplementary material.*

1 INTRODUCTION

Image generation has witnessed remarkable progress in recent years, powered by advances in generative models such as generative adversarial networks (GANs) Krichen (2023); Mirza & Osindero (2014); Isola et al. (2017) and diffusion models Ho et al. (2020); Song et al. (2021). These techniques have allowed a wide range of applications in creative design Ruiz et al. (2023), large image synthesis Le et al. (2024) and story visualization Wu et al. (2024). In addition to improving image quality, many works Zhang et al. (2023); Mou et al. (2023); Li et al. (2024; 2025) have explored various input conditions to introduce more control to the generation process, including semantic layouts, sketches, or pose keypoints. These advances reflect the community's growing attention to generating object images that are not only high-quality but also easily manipulable.

Despite recent progress in controllable image generation, producing object images under strong structural constraints remains a significant challenge, particularly when precise control over an object's topology and spatial layout is required. An illustration of these structural constraints is shown in Fig. 1. This level of control is essential because the structural configuration of an object, such as how its parts are arranged and connected, plays a central role in defining its category, semantics and functionality. And fine-grained structural control in image generation is crucial for applications such as multiview-consistent image generation Liu et al. (2023b), 3D asset design Tang (2022), and robotics domain Tremblay et al. (2018a;b), where object structure must be explicitly specified and faithfully preserved. Moreover, performing appearance-consistent structural editing on generated images is also important, such as refining specific components or generating articulated objects under different joint states FANG et al. (2024).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

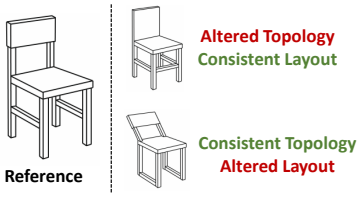

Figure 1: Illustration of an object's topology and spatial layout using chair as an example.

To generate photorealistic object images under topology and spatial layout constraints and support structural editing, two key challenges need to be addressed. 1) **How to enable users to express and revise their structural intent in an accurate, flexible, and user-friendly manner**. Specifically, topology and spatial layout constraints are difficult to express through conventional input conditions, such as text, semantic layouts, or pose keypoints, which typically provide only coarse-grained guidance of structural constraints. More fine-grained conditions, such as sketches, while offering greater structural precision, require drawing expertise and precise spatial reasoning, making them difficult for non-expert users and inconvenient for viewpoint or structural modification. 2) **How to faithfully translate the structural intent into photorealistic images.** Most existing controllable generation methods such as ControlNet Zhang et al. (2023); Li et al. (2024) and LoRA Hu et al. (2022); Dettmers et al. (2023) often perform semantic-level or pixel-level alignment, which hinders them from generating natural geometric details and smooth connections between primitives, making it difficult to simultaneously ensure realism and structural fidelity in the generated images.

To address these challenges, we propose **Struct2Real**, a novel framework that generates photorealistic object images while preserving topology and spatial layout, and supports flexible viewpoint and structural editing. Our method is built on two core designs. **First, we introduce an effective 3D representation for object structure to the image generation task and design a structure modeling system around it.** This 3D representation encodes the topology and spatial layout of an object using a composition of geometric primitives. We refer to this representation as StructMap. This system ensures that users can express and revise their structural intent accurately, flexibly, and conveniently, providing an expressive prior for controllable image generation. **Second, we develop the Structure-Aware Generation Agent (SAGA), a skill-oriented agent that enables photorealistic object image generation and editing under structural constraints.** SAGA decomposes the generation procedure into specialized skills designed to enhance photorealism and structural consistency, and adaptively coordinates them under the constraints encoded in StructMap. This formulation enables high-quality image synthesis with strong structural fidelity, even in scenarios involving complex topologies and fine-grained part relations. Moreover, SAGA integrates a realism memory module that retains visual attributes throughout generation, enabling appearance-consistent viewpoint editing and structural editing. Examples of StructMaps and corresponding generated images are shown in Fig. 2.

We compare Struct2Real against state-of-the-art baselines on generating photorealistic object images under topology and spatial layout constraints. The comparison covers multiple evaluation aspects, including image realism, structural alignment, and condition accessibility. The results demonstrate that Struct2Real achieves superior performance in generating photorealistic and structurally faithful images at the cost of reasonable user effort.

In summary, this work makes the following key contributions: 1) In the task of structure-grounded object image generation, we introduce an effective 3D representation called StructMap and develop a StructMap-driven structure modeling system, enabling users to express and revise their structural intent accurately and conveniently. 2) We introduce SAGA, an agent that formulates structure-grounded image synthesis as a dynamic skill orchestration process and incorporates a realism memory module, enabling photorealistic object image generation conditioned on StructMap and supporting viewpoint or structural editing. 3) We conduct comprehensive experiments and provide in-depth analysis to study the effectiveness of Struct2Real, and the results demonstrate its superiority.

2 Related Works

Controllable Image Generation. Text-to-image generation has made significant progress Nichol et al. (2022); Ramesh et al. (2022); Saharia et al. (2022), largely driven by diffusion models Ho et al. (2020); Song et al. (2021). To provide more explicit control, prior works have introduced complementary conditions, including bounding boxes Cheng et al. (2023); Jia et al. (2024a); Qu et al. (2023); Xie et al. (2023); Yang et al. (2023a); Zheng et al. (2023), segmentation maps Avrahami et al. (2023); Bar-Tal et al. (2023); Couairon et al. (2023); Gafni et al. (2022), sketches Bashkirova et al. (2023); Wang et al. (2024a); Koley et al. (2024), and depth maps Lee et al. (2024); recent



Figure 2: Visualizations of Struct2Real generation results, including direct generation, viewpoint editing and structural editing. For each sub-column, the top row presents a StructMap input and the bottom row shows the corresponding photorealistic image generated by SAGA.

methods further unify multiple input modalities into flexible control interfaces Wang et al. (2024b); Sun et al. (2024). More closely related to our setting, (Vavilala et al., 2023) represents scenes using editable 3D primitives and trains a conditioned generator on their rendered projections, while (Bhat et al., 2023) enables controllable synthesis from generalized depth and editable 3D boxes. These works demonstrate the effectiveness of simplified 3D scaffolds for conditioned realistic synthesis. In contrast, our work focuses on object-level structural specification, where StructMap explicitly represents object parts and their topology, connectivity, relative pose, and spatial layout, and couples this representation with a structure-aware generation process for faithful realization and editing.

MLLM and Agent for Image Generation. LLMs and MLLMs have increasingly been used to enhance image generation, including generating spatial layouts Feng et al. (2023); Lian et al. (2024); Yang et al. (2024); Wang et al. (2024d), serving as enhanced text encoders Feng et al. (2024); Tan et al. (2024); Liu et al. (2024); Xia et al. (2024); Song et al. (2024), and improving generation of rare concepts or global consistency Park et al. (2025); Kwon et al. (2024). Several works further employ LLM-based agents to orchestrate visual models or generation tools for iterative synthesis and refinement Wu et al. (2023a); Jiang et al. (2026); Qin et al. (2024); Wang et al. (2024c); Zhao et al. (2024); Jia et al. (2024b). Particularly relevant are (Yang et al., 2023b), which performs multimodal iterative self-refinement through prompt revision, draft selection, feedback, and memory, and (Wu et al., 2023b), which introduces an LLM-controlled generate–assess–correct loop for text-to-image generation. These works establish iterative agentic refinement as an effective paradigm for image synthesis. Our SAGA builds on this general paradigm but specializes it for structure-grounded object generation by explicitly diagnosing structural inconsistency, insufficient geometric detail, and appearance defects, and invoking corresponding structure-aware skills. Its realism memory further stores object-specific visual attributes to support appearance-consistent viewpoint and structural editing.

3 METHOD

3.1 OVERVIEW

Our goal is to generate object images under structural control, ensuring that the outputs not only appear natural, realistic, and detail-rich, but also satisfy topology and spatial layout constraints.

Formally, we need to find a structural condition S_M that encodes the object's topology and spatial layout S, and then produce an image $I = \mathcal{F}(S_M)$, where $\mathcal{F}$ denotes a generative process. The generated image I is expected to possess two key properties: (i) **Structural Faithfulness**: preserving the structural configurations defined by S_M. (ii) **Visual Realism**: appearing nearly indistinguishable from real-world objects, as if captured in actual photographs. Moreover, the generative process $\mathcal{F}$ is expected to support viewpoint and structural editing by producing an updated image $I' = \mathcal{F}(S'_M)$ under a modified structural condition S'_M, while maintaining consistent visual attributes across edits.

To this end, we present Struct2Real, a framework with two main modules: 1) a structure modeling system that allows users to conveniently create a novel structural condition S_M, called StructMap, which accurately encodes the topology and spatial layout of an object. 2) the Structure-Aware Generation Agent (SAGA), a skill-oriented agent that realizes the structure-conditioned process $\mathcal{F}$, producing photorealistic images while preserving structural configurations defined by S_M and enabling viewpoint and structural editing. In the following sections, we introduce these components and the generation process in detail.

3.2 StructMap and Structure Modeling System

While previous methods for controllable image generation have explored a variety of input conditions such as text, semantic layouts, or pose keypoints, these conditions are often coarse-grained or ambiguous, making it difficult to accurately reflect the object's topology and spatial layout. And more fine-grained conditions, such as sketches, are often difficult for non-expert users to create and inconvenient for viewpoint and structural modification, as accurately depicting and revising object structures typically requires both artistic expertise and strong spatial understanding. In comparison, we develop a structure modeling system that enables users to create an effective 3D structural representation named StructMap. It serves as a prior to guide the generation process with explicit control over object topology and spatial layout. Building upon StructMap, we develop a part-based workflow embedded in an interactive design interface allowing users to conveniently organize geometric primitives into StructMaps with fully customizable topology and spatial layout. An illustration of the structure modeling system is shown in Fig. 3.

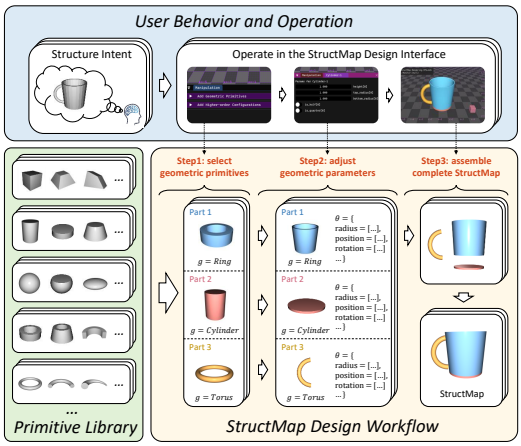

Figure 3: Illustration of the structure modeling system. The user creates the desired StructMap according to their structural intent using our StructMap design workflow and interface.

Definition of StructMap. We define a **StructMap** as an explicit 3D representation encoding an object's topology and spatial layout, composed of a set of geometric primitives and their spatial relationships. Formally, a StructMap is represented as a primitive set $\{(g_i, \theta_i)\}_{i=1}^{N}$, where each $g_i \in \mathcal{G}$ is a geometric primitive selected from a primitive library $\mathcal{G}$, and θ_i is a set of interpretable parameters that determine the primitive's shape, position, and orientation in 3D space. Each primitive (g_i, θ_i) serves as a component or "part" of the object. The topology of the object is revealed by the contact relationships of these parts, while the spatial layout is encoded by their poses in 3D space.

Motivation and Rationale Behind StructMap. Our goal is to develop a 3D representation that can accurately encode an object's topology and spatial layout while remaining easy to create and revise. While point-line representations Haridis (2020); Xu et al. (2025) provide a simple way to describe structural connectivity, they lack 3D volume and thus cannot explicitly represent part size, shape, or precise attachment regions. We therefore model each basic component as a controllable 3D geometric primitive, which enables finer-grained specification of part geometry, spatial relations, and connectivity. This representation is extensible to broader object categories by introducing new primitives and parameterizations, and is also consistent with the Recognition-by-Components theory Biederman (1987), which suggests that humans perceive objects through simple geometric components and their spatial configurations. *More detailed discussions are provided in Appendix A.1.*

StructMap Design Workflow and Interface. We develop a part-based workflow tailored for designing StructMaps. This workflow defines how users can organize geometric primitives into meaningful structural configurations. First, users select and instantiate geometric primitives from the primitive library according to their design requirements. Since each primitive can be adjusted via interpretable parameters such as size, position, and orientation, users can modify them to achieve the desired geometry and pose. Once all primitives have been composed and configured, the user completes the construction of a StructMap that explicitly encodes their intended structure. To support

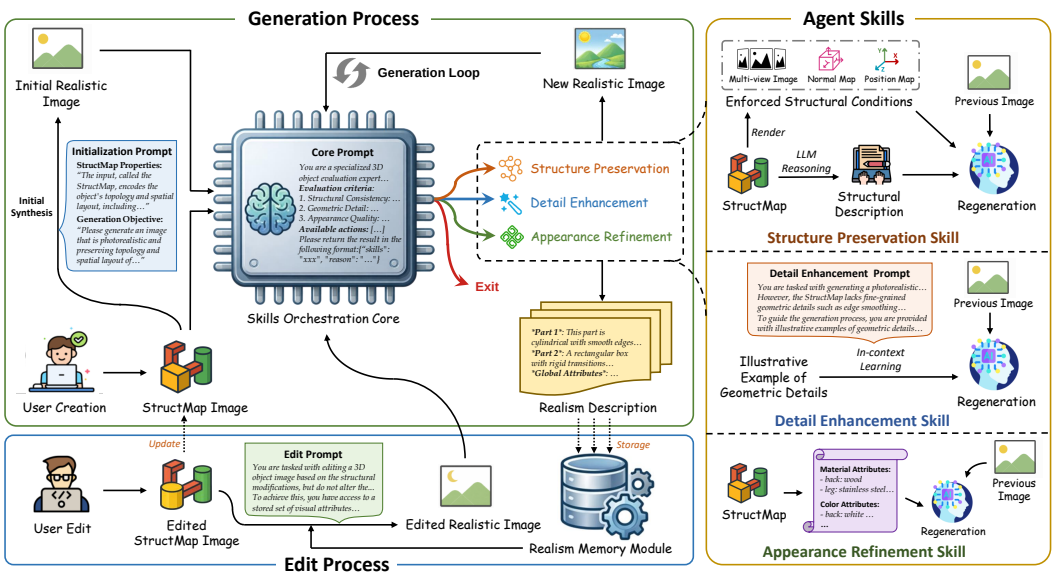

Figure 4: Illustration of the Structure-Aware Generation Agent.

this workflow and facilitate practical usage, we implement an interactive design interface that enables users to build StructMaps. In this interface, we design several high-order configurations based on common object structures, enabling users to instantiate multiple primitives simultaneously. *Examples of the high-order configurations are provided in Appendix A.8.* And the interface allows users to reuse previously created StructMaps as components when creating new ones. These designs further accelerate the creation of StructMaps. We include screenshots of our interface in Fig. 3. *A more detailed video demonstration of our workflow and interface is provided in the supplementary material* `design_workflow_demonstration.mp4`. Once a 3D StructMap is created, we render it as a 2D image from one or more viewpoints, forming a structural visual condition for the subsequent generation. In the meantime, the 3D StructMap can be exported and reused in case of further editing.

Discussions. 1) Compared to sketches, StructMap is natively 3D and encodes 3D attributes such as depth and symmetry, while also lowering the skill barrier for users and making viewpoint and structural edits easy to perform. 2) Unlike CAD, which involves precise 3D modeling and requires professional expertise, our system is purpose-built for structure modeling, offering a more accessible and user-friendly alternative. *Please refer to Appendix A.1 for more detailed discussions.*

3.3 Structure-Aware Generation Agent

After obtaining a StructMap that encodes the user's structural intent, we need a generative process to leverage the comprehensive structural information in the StructMap and generate photorealistic object images while preserving the structural configuration it encodes. Multimodal large language models (MLLMs) have demonstrated strong capabilities in understanding structural and semantic information in images and generating high-quality visual content. However, due to the inherent complexity and task-specific demands of structure-grounded image generation, directly applying MLLMs to synthesize realistic images often results in issues related to structural consistency and visual realism. Motivated by the paradigm of agent skills, we develop the Structure-Aware Generation Agent (SAGA), which decomposes the key objectives in structure-grounded image synthesis into specialized generation skills and formulates the generation process as a dynamic skill orchestration procedure. Furthermore, to support appearance-consistent viewpoint editing and structural editing, SAGA incorporates a realism memory module that retains salient visual attributes throughout the generation process. The overall workflow of SAGA is illustrated in Fig. 4.

Overall Generation Pipeline. Given a StructMap image as the structural condition, SAGA first performs an initial synthesis step. During this stage, an initialization prompt provides the agent with the fundamental task specification by (i) clarifying that the StructMap encodes object topology and spatial layout while omitting surface-level appearance, and (ii) defining the generation objective as synthesizing a photorealistic image that faithfully preserves the structural configurations encoded in StructMap. Subsequently, the StructMap image and the initially generated image are jointly fed

into the skill orchestration core, which evaluates the current generated image in terms of structural consistency and visual realism. Based on this assessment, the core automatically selects an appropriate generation skill module to refine the image. This evaluation, skill selection, and refinement procedure is performed iteratively until the orchestration core determines that the image satisfies structural and realism criteria, or a predefined maximum number of iterations is reached.

Agent Skills. To fulfill the objectives of structure-grounded image synthesis, we design three specialized generation skills, each targeting a distinct aspect of the task:

- **Structure Preservation Skill.** This skill is activated when structural inconsistencies are detected between the generated image and the StructMap. It evaluates topology and spatial layout, and enforces stronger structural constraints to correct structural drift. Specifically, to supply sufficient geometric and spatial information, we first provide a set of enforced structural conditions including multi-view renderings, normal maps and position maps of the StructMap (these conditions are not provided from the beginning to reduce unnecessary overhead). Then, to reduce the modality gap between visual inputs and the language-oriented reasoning capabilities of MLLMs Liu et al. (2023a), we augment these structural conditions with a textual structural description, which encodes the topology and spatial layout of the StructMap, including the number of parts, their spatial arrangement, and connectivity relationships. Finally, the agent integrates all visual and textual structural information to regenerate the image based on the previous generated image.
- **Detail Enhancement Skill.** This skill is triggered when the generated image lacks sufficient geometric details. It supplements missing geometric details, such as smoothing sharp geometric edges, refining transitional connections between parts, and enriching surface patterns, to bridge the realism gap between the StructMap and real-world objects. Considering that the concept of geometric details is ambiguous and may not be reliably interpreted by MLLMs, we adopt an in-context learning strategy by providing carefully curated illustrative examples of geometric details. These examples guide the agent to synthesize images with richer geometric complexity.
- **Appearance Refinement Skill.** This skill is employed when the image exhibits deficiencies in texture and material realism. It enhances texture coherence, material plausibility and overall aesthetic. We first prompt the agent to infer plausible texture and material attributes for different object parts based on the object category, component functions, and their geometric characteristics. These inferred appearance attributes are then used as guidance to regenerate the image with improved visual plausibility and aesthetic.

Realism Memory Mechanism. SAGA is designed to support viewpoint and structural editing with consistent visual attributes, which means when the StructMap image specifies changes in viewpoint or structure, the generated image is updated accordingly while preserving previously established geometric details and texture-material attributes. To this end, we require SAGA to produce a realism description alongside each generated image, explicitly documenting the geometric details and texture-material attributes introduced during synthesis. These descriptions are stored in a dedicated *Realism Memory Module*, which serves as a persistent repository of stable visual attributes. When users perform viewpoint or structural editing, the stored visual attributes are incorporated as guidance, ensuring that modifications are carried out while preserving the geometric details and texture-material attributes, thereby achieving appearance-consistent structural and viewpoint editing.

4 Experiments

4.1 Experiment Setup

Data Preparation. Existing datasets for controllable image generation are often constructed by extracting conditioning inputs from pre-existing images Lin et al. (2015). While this approach is convenient, it limits the input conditions to those derivable from existing images and thereby restricts user's creativity. Moreover, these datasets typically lack alignment across different conditions, making it difficult to evaluate structure-grounded generation in a comparable way. To address these limitations, we construct a structure-prior dataset comprising diverse, manually created inputs across multiple conditions, enabling unrestricted condition creation and cross-condition comparison. We focus on four conditioning inputs that can be flexibly created from scratch by users: textual descriptions, lineart, scribbles, and our proposed StructMap. To ensure fair comparisons across different conditions, we require that each set of input conditions describe the same object structure. Formally, for each structure S, we collect a tuple (T, L, S_C, S_M) where T, L, S_C, S_M denote text, lineart, scribble and

Table 1: **[Left]** Quantitative comparison across different conditions and approaches. **[Right]** Average creation time and creation difficulty (1-5 scale) for each condition.

Condition	Method	FID↓	SSIM↑	LPIPS↓	MOS-R↑	MOS-A↑	Creation Time (min)	Creation Difficulty
Text	OmniGen	43.17	–	–	1.73	1.25	2.79	2.09
	Lumina-Image	43.49	–	–	1.67	1.42		
	GPT Image 2	39.24	–	–	4.33	1.92		
Lineart	ControlNet++	41.56	**0.706**	0.097	2.67	4.62	25.76	4.39
	ControlAR	49.97	0.136	0.832	1.61	1.89		
	GPT Image 2	38.74	0.647	1.140	4.58	4.51		
Scribble	ControlNet	40.37	0.492	0.302	2.88	3.51	3.26	2.26
	T2I-Adapter	42.97	0.662	0.113	2.53	3.87		
	GPT Image 2	38.91	0.607	0.134	4.19	4.05		
StructMap	Struct2Real	**38.51**	0.694	**0.091**	**4.75**	**4.71**	6.69	2.68

StructMap respectively, and all aimed at representing the same S. Overall, our dataset covers 30 object categories and includes a total of 3000 samples. *More details on dataset construction and the diversity of StructMap structures in the dataset are provided in Appendix A.2.1 and A.10.*

Baseline Methods. For each condition, we select two models that are either widely adopted or represent state-of-the-art advancements under that condition. Specifically, we compare against OmniGen Xiao et al. (2024) and Lumina-Image 2.0 Qin et al. (2025) for text condition, ControlNet++ Li et al. (2024) and ControlAR Li et al. (2025) for lineart condition, and ControlNet Zhang et al. (2023) and T2I-Adapter Mou et al. (2023) for scribble condition. In addition, we additionally compare our method against MLLM-based image generation under these conditions.

Evaluation Metrics. Our task is to generate photorealistic object images under topology and spatial layout constraints. Accordingly, we need to evaluate our results from two perspectives: image realism and structural fidelity. For image realism, we use the CLIP-based FID, and for structural fidelity, we employ SSIM and LPIPS. In addition, we conduct a two-part Mean Opinion Score (MOS) study Huynh-Thu et al. (2011), where human participants evaluate (1) the realism of generated images (**MOS-R**) and (2) the alignment between the generated image and the input structure (**MOS-A**) on a 1-5 Likert scale Likert (1932). Notably, for MOS-A evaluation, we instruct human participants to focus primarily on the object's topology and spatial layout, rather than merely assessing contour-level alignment. *The scoring criteria of the human evaluation are provided in Appendix A.2.5.* Additionally, we perform a fine-grained structural evaluation along five dimensions; see Appendix A.7.

Choice of MLLMs. SAGA is compatible with various MLLMs as the underlying model for each component, so we conducted experiments to assess the performance of different MLLMs in our task. Among them, GPT-Image-2 (OpenAI, 2026) achieved the best performance (*detailed comparisons provided in Appendix A.5.1*). Therefore, all experiments presented in this section are conducted using GPT-Image-2 as the underlying model.

4.2 Results & Analysis

We compare the performance of our approach against other baseline methods. Related results are shown in Tab. 1-Left and Fig. 5.

Realism of the Generated Images. As shown in Tab. 1-Left, our method achieves the best performance on both FID scores and MOS-R ratings. We attribute this superiority to two key factors: (1) StructMap provides clear and straightforward structural guidance, and (2) SAGA effectively generates realistic details while faithfully preserving the specified structure. In contrast, other methods achieve worse performance on both FID scores and MOS-R ratings. Textual prompts often suffer from ambiguity or under-specification, while methods based on lineart and scribble tend to over-emphasize edge alignment, sometimes at the cost of realism and natural visual appearance. Further, as shown in Fig. 5, our method generates object images with significantly better visual realism. The generated objects appear indistinguishable from real-world objects at first glance. In contrast, other methods may produce visual flaws, such as unnatural textures (Lineart & Controlnet++ result in example (a)) or distorted shapes (Scribble & Controlnet result in example (b)), which make the generated images appear synthetic. Additionally, when we replaced the image generation model with GPT Image 2, the realism of the results improved; however, some irregular geometric shapes (Text & GPT Image 2 in example (b)) and insufficiently rich geometric details and texture (Lineart & GPT Image 2 in example (a)) still appeared, making the generated objects look less realistic.

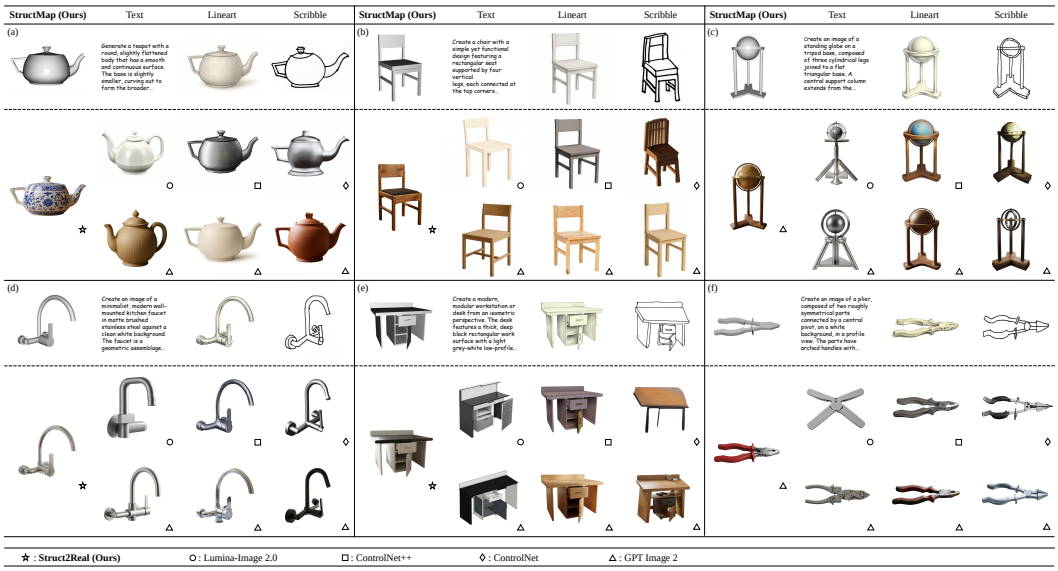

Figure 5: Qualitative results across different baseline methods. For each sub-panel, the top row shows the input conditions for each method, and the bottom two rows show the object images generated by different methods. The icon in the bottom-right corner of each image indicates the method used to generate it, and the meaning of each icon is explained in the legend at the bottom. *Due to space limitation, more results and full textual descriptions are provided in Appendix A.3.*

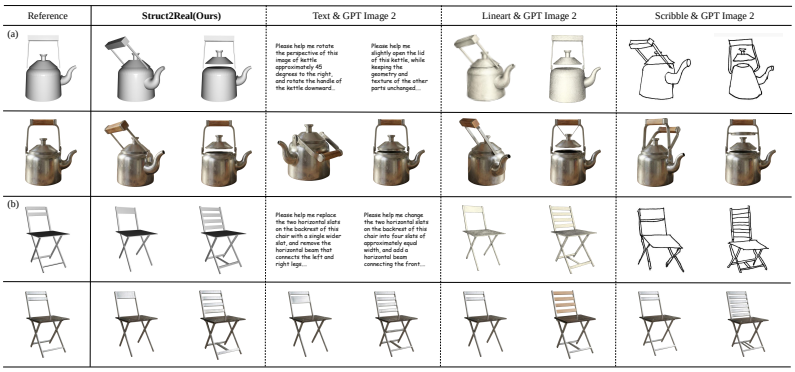

Figure 6: Qualitative results of the viewpoint and structural editing experiment.

Structural Alignment with the Conditioning Input. From Fig. 5, we observe that text and scribble conditioned generation yields weak structural alignment. This indicates that such conditions are less effective to convey precise structural constraints to the model, making it difficult to enforce accurate object structure. The image generated by Lineart & Controlnet++ strongly preserves the structural constraints encoded in the condition. However, this strength can be a double-edged sword: image generation models may overfit to low-level features(Lineart & Controlnet++ in example (b)), reproducing input noise and prioritizing pixel consistency at the expense of realism. In comparison, our method consistently preserves the underlying topology and spatial layout. This is further supported by SSIM, LPIPS and MOS-A scores in Tab. 1-Left, where our results perform much better than text and scribble and comparably to lineart, indicating that the structural fidelity is effectively preserved in human visual perception. Additionally, when we replaced the image generation model with GPT Image 2, structural errors occasionally persisted (Lineart & GPT Image 2 in example (d) and Scribble & GPT Image 2 in example (e)).

Accessibility of Different Conditions. We also analyze the accessibility of each condition using creation time and creation difficulty collected during dataset construction. Specifically, we 1) record the average time consumed to create each type of condition during the creation process, and 2) ask each creator to rate the difficulty of producing data of each type on a 1–5 Likert scale based on their perceived mental effort and the level of artistic skill required. As indicated by the results in Tab. 1-Right, lineart, despite its proficiency in guiding image generation, is considered the most complicated

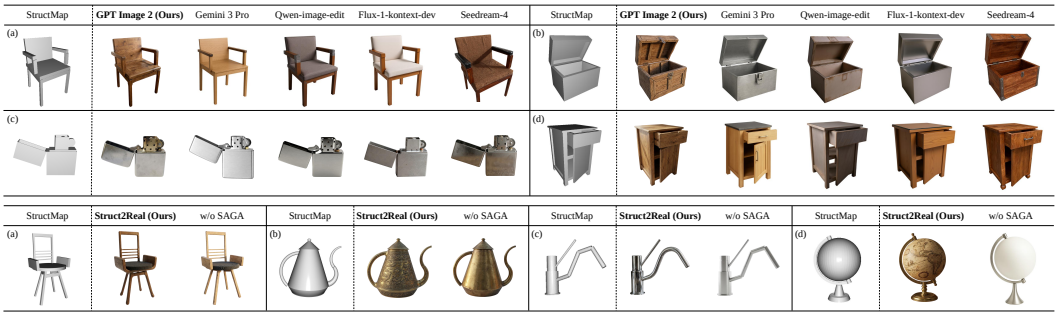

Figure 7: **[Top]** Ablation study results across different underlying MLLMs. **[Bottom]** Ablation study results using SAGA or MLLM alone.

to produce among the four types. This significantly adds to the difficulties in the acquisition of high quality generated images. In comparison, StructMap is rated with a much less creation time and difficulty, making it the optimal choice of conditioning in controllable image generation.

4.3 Viewpoint and Structural Editing

To evaluate the effectiveness of Struct2Real in viewpoint and structural editing, we conduct an editing experiment. Specifically, we select a subset of objects from our constructed dataset and alter the viewpoints and structures of their StructMap images. The images are then regenerated using SAGA, conditioned on both the updated StructMap images and the previously generated images. For comparison, we also describe the same modifications using text, lineart, and scribbles, and perform image generation with GPT Image 2. As shown in Figure 6, the images generated by Struct2Real faithfully reflect the intended viewpoint and structural edits, while ensuring that the regenerated images maintain consistent geometric details and texture-material attributes with the original images. This can be attributed to (i) the StructMap's ability to accurately represent viewpoint and structural edits, as well as (ii) the design of SAGA's skills and realism memory, which together ensure consistency in both structure and appearance. In contrast, images generated using alternative input conditions with GPT exhibit misaligned viewpoints(Lineart & GPT in example (a)-[Left]), structural inconsistencies(Text & GPT in example (b)), and appearance discrepancies(Lineart & GPT in example (b)-[Right]). *Detailed results and analysis are provided in Appendix A.4.*

4.4 Ablation Study

To validate the design of Struct2Real, we conduct a series of ablation studies. Specifically, we first assess the generality of SAGA by replacing the underlying model in SAGA with other MLLMs. Second, we validate the effectiveness of SAGA by comparing the results generated by SAGA to those generated by directly employing an MLLM (both conditioned on StructMap). Some experimental results are presented in Fig. 7. As shown in Fig. 7-top, SAGA achieves consistently strong results when applied with all underlying MLLMs. In Fig. 7 (bottom), we observe that images generated by our method exhibit higher structural fidelity (example (a)), richer geometric details (example (b) and (c)), and more realistic texture and material attributes (example (d)). Furthermore, we evaluate the effectiveness of StructMap by applying SAGA under alternative input conditions, including text, line art, and scribbles, and verify the contribution of each component in SAGA by removing the corresponding module. *Detailed results and analysis are provided in Appendix A.5.*

5 Conclusion

We propose Struct2Real, a novel framework for photorealistic object image generation under structural control. Central to our method is StructMap, an explicit 3D representation that encodes object topology and spatial layout via geometric primitives. Around StructMap, we design a structure modeling system, enabling users to conveniently design StructMaps. Then, we develop SAGA to generate photorealistic images that faithfully preserve the topology and spatial layout of the input structure. Comprehensive experiments show that Struct2Real achieves high structural fidelity and visual quality with relatively low user effort. Struct2Real offers a new perspective on controllable image generation by introducing symbolic structural design.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing of the manuscript. Generative AI models are also used as explicit components of the proposed method and experimental baselines, as described in the main paper; these uses are part of the evaluated systems rather than undisclosed authoring assistance. We did not rely on generative AI to independently determine experimental results or scientific conclusions. All AI-assisted content was manually reviewed, revised, and verified by the authors. In particular, factual claims and literature references were checked against their original sources, and experimental results were obtained and validated through the evaluation procedures described in the paper. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18370–18380, 2023. doi: 10.1109/CVPR52729.2023.01762.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation, 2023. URL https://arxiv.org/abs/2302.08113.

Dina Bashkirova, Jose Lezama, Kihyuk Sohn, Kate Saenko, and Irfan Essa. Masksketch: Unpaired structure-guided masked image generation, 2023. URL https://arxiv.org/abs/2302.05496.

Shariq Farooq Bhat, Niloy J. Mitra, and Peter Wonka. Loosecontrol: Lifting controlnet for generalized depth conditioning, 2023.

Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

Jiaxin Cheng, Xiao Liang, Xingjian Shi, Tong He, Tianjun Xiao, and Mu Li. Layoutdiffuse: Adapting foundational diffusion models for layout-to-image generation, 2023. URL https://arxiv.org/abs/2302.08908.

Guillaume Couairon, Marlène Careil, Matthieu Cord, Stéphane Lathuilière, and Jakob Verbeek. Zero-shot spatial layout conditioning for text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2174–2183, 2023. doi: 10.1109/ICCV51070.2023.00207.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

Zhou FANG, Yong-Lu Li, Lixin Yang, and Cewu Lu. General articulated objects manipulation in real images via part-aware diffusion process. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=WRd9LCbvxN.

Weixi Feng, Wanrong Zhu, Tsu-Jui Fu, Varun Jampani, Arjun Reddy Akula, Xuehai He, S Basu, Xin Eric Wang, and William Yang Wang. LayoutGPT: Compositional visual planning and generation with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Xu8aG5Q8M3.

Yutong Feng, Biao Gong, Di Chen, Yujun Shen, Yu Liu, and Jingren Zhou. Ranni: Taming text-to-image diffusion for accurate instruction following. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4744–4753, June 2024.

Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors, 2022. URL https://arxiv.org/abs/2203.13131.

Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xuanda Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Xin Xia, Xuefeng Xiao, Zhonghua Zhai, Xinyu Zhang, Qi Zhang, Yuwei Zhang, Shijia Zhao, Jianchao Yang, and Weilin Huang. Seedream 3.0 technical report, 2025. URL https://arxiv.org/abs/2504.11346.

Alexandros Haridis. The topology of shapes made with points. *CoRR*, abs/2008.05262, 2020. URL https://arxiv.org/abs/2008.05262.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Quan Huynh-Thu, Marie-Neige Garcia, Filippo Speranza, Philip Corriveau, and Alexander Raake. Study of rating scales for subjective quality assessment of high-definition video. *IEEE Transactions on Broadcasting*, 57(1):1–14, 2011. doi: 10.1109/TBC.2010.2086750.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

Chengyou Jia, Minnan Luo, Zhuohang Dang, Guang Dai, Xiaojun Chang, Mengmeng Wang, and Jingdong Wang. Ssmg: Spatial-semantic map guided diffusion model for free-form layout-to-image generation, 2024a. URL https://arxiv.org/abs/2308.10156.

Chengyou Jia, Changliang Xia, Zhuohang Dang, Weijia Wu, Hangwei Qian, and Minnan Luo. Chatgen: Automatic text-to-image generation from freestyle chatting, 2024b. URL https://arxiv.org/abs/2411.17176.

Kaixun Jiang, Yuzheng Wang, Junjie Zhou, Pandeng Li, Zhihang Liu, Chen-Wei Xie, Zhaoyu Chen, Yun Zheng, and Wenqiang Zhang. Genagent: Scaling text-to-image generation via agentic multimodal reasoning, 2026. URL https://arxiv.org/abs/2601.18543.

Subhadeep Koley, Ayan Kumar Bhunia, Deeptanshu Sekhri, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. It's all about your sketch: Democratising sketch control in diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7204–7214, 2024. doi: 10.1109/CVPR52733.2024.00688.

Moez Krichen. Generative adversarial networks. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–7, 2023. doi: 10.1109/ICCCNT56998.2023.10306417.

Soyeong Kwon, Taegyeong Lee, and Taehwan Kim. Zero-shot text-guided infinite image synthesis with llm guidance, 2024. URL https://arxiv.org/abs/2407.12642.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Minh-Quan Le, Alexandros Graikos, Srikar Yellapragada, Rajarsi Gupta, Joel Saltz, and Dimitris Samaras. ∞-brush: Controllable large image synthesis with diffusion models in infinite dimensions, 2024. URL https://arxiv.org/abs/2407.14709.

Jonghyun Lee, Hansam Cho, YoungJoon Yoo, Seoung Bum Kim, and Yonghyun Jeong. Compose and conquer: Diffusion-based 3d depth aware composable image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=p4eG8rCa0b.

Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. Controlnet++: Improving conditional controls with efficient consistency feedback. In *European Conference on Computer Vision (ECCV)*, 2024.

Zongming Li, Tianheng Cheng, Shoufa Chen, Peize Sun, Haocheng Shen, Longjin Ran, Xiaoxin Chen, Wenyu Liu, and Xinggang Wang. Controlar: Controllable image generation with autoregressive models. In *International Conference on Learning Representations*, 2025.

Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. LLM-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=hFALpTb4fR. Featured Certification.

Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):1–55, 1932.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Mushui Liu, Yuhang Ma, Yang Zhen, Jun Dan, Yunlong Yu, Zeng Zhao, Zhipeng Hu, Bai Liu, and Changjie Fan. Llm4gen: Leveraging semantic representation of llms for text-to-image generation, 2024. URL https://arxiv.org/abs/2407.00737.

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023b.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL http://arxiv.org/abs/1411.1784.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

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, 2022. URL https://arxiv.org/abs/2112.10741.

OpenAI. GPT-Image-2, 2026. URL https://developers.openai.com/api/docs/models/gpt-image-2. OpenAI API model documentation, accessed September 20, 2026.

Dongmin Park, Sebin Kim, Taehong Moon, Minkyu Kim, Kangwook Lee, and Jaewoong Cho. Rare-to-frequent: Unlocking compositional generation power of diffusion models on rare concepts with LLM guidance. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=BgxsmpVoOX.

Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. A new era of intelligence with gemini 3. https://blog.google/products-and-platforms/products/gemini/gemini-3/, November 2025. Published November 18, 2025.

Jie Qin, Jie Wu, Weifeng Chen, Yuxi Ren, Huixia Li, Hefeng Wu, Xuefeng Xiao, Rui Wang, and Shilei Wen. Diffusiongpt: Llm-driven text-to-image generation system. *CoRR*, abs/2401.10061, 2024. URL https://doi.org/10.48550/arXiv.2401.10061.

Qi Qin, Le Zhuo, Yi Xin, Ruoyi Du, Zhen Li, Bin Fu, Yiting Lu, Xinyue Li, Dongyang Liu, Xiangyang Zhu, Will Beddow, Erwann Millon, Wenhai Wang Victor Perez, Yu Qiao, Bo Zhang, Xiaohong Liu, Hongsheng Li, Chang Xu, and Peng Gao. Lumina-image 2.0: A unified and efficient image generative framework, 2025. URL https://arxiv.org/pdf/2503.21758.

Leigang Qu, Shengqiong Wu, Hao Fei, Liqiang Nie, and Tat-Seng Chua. Layoutllm-t2i: Eliciting layout guidance from llm for text-to-image generation, 2023. URL https://arxiv.org/abs/2308.05095.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. URL https://arxiv.org/abs/2204.06125.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=08Yk-n5l2Al.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Kunpeng Song, Yizhe Zhu, Bingchen Liu, Qing Yan, Ahmed Elgammal, and Xiao Yang. Moma: Multimodal llm adapter for fast personalized image generation, 2024. URL https://arxiv.org/abs/2404.05674.

Yanan Sun, Yanchen Liu, Yinhao Tang, Wenjie Pei, and Kai Chen. Anycontrol: Create your artwork with versatile control on text-to-image generation, 2024.

Zhiyu Tan, Mengping Yang, Luozheng Qin, Hao Yang, Ye Qian, Qiang Zhou, Cheng Zhang, and Hao Li. An empirical study and analysis of text-to-image generation using large language model-powered textual representation, 2024. URL https://arxiv.org/abs/2405.12914.

Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. https://github.com/ashawkey/stable-dreamfusion.

Tencent Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025.

Jonathan Tremblay, Thang To, and Stan Birchfield. Falling things: A synthetic dataset for 3d object detection and pose estimation. *CoRR*, abs/1804.06534, 2018a. URL http://arxiv.org/abs/1804.06534.

Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. In *Conference on Robot Learning (CoRL)*, 2018b. URL https://arxiv.org/abs/1809.10790.

Vaibhav Vavilala, Seemandhar Jain, Rahul Vasanth, Anand Bhattad, and David Forsyth. Blocks2world: Controlling realistic scenes with editable primitives, 2023. URL https://arxiv.org/abs/2307.03847.

Xi Wang, Hongzhen Li, Heng Fang, Yichen Peng, Haoran Xie, Xi Yang, and Chuntao Li. Lineart: A knowledge-guided training-free high-quality appearance transfer for design drawing with diffusion model, 2024a. URL https://arxiv.org/abs/2412.11519.

Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation, 2024b. URL https://arxiv.org/abs/2402.03290.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 128374–128395. Curran Associates, Inc., 2024c. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e7c786024ca718f2487712bfe9f51030-Paper-Conference.pdf.

Zhenyu Wang, Enze Xie, Aoxue Li, Zhongdao Wang, Xihui Liu, and Zhenguo Li. Divide and conquer: Language models can plan and self-correct for compositional text-to-image generation, 2024d. URL https://arxiv.org/abs/2401.15688.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models, 2023a. URL https://arxiv.org/abs/2303.04671.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

Jianzong Wu, Chao Tang, Jingbo Wang, Yanhong Zeng, Xiangtai Li, and Yunhai Tong. Diffsensei: Bridging multi-modal llms and diffusion models for customized manga generation. *arXiv preprint arXiv:2412.07589*, 2024.

Tsung-Han Wu, Long Lian, Joseph E Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting llm-controlled diffusion models. *arXiv preprint arXiv:2311.16090*, 2023b.

Bin Xia, Shiyin Wang, Yingfan Tao, Yitong Wang, and Jiaya Jia. Llmga: Multimodal large language model based generation assistant, 2024. URL https://arxiv.org/abs/2311.16500.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340*, 2024.

Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7452–7461, 2023.

Yuanyou Xu, Zongxin Yang, and Yi Yang. Skdream: Controllable multi-view and 3d generation with arbitrary skeletons. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 314–325, 2025. doi: 10.1109/CVPR52734.2025.00038.

Binbin Yang, Yi Luo, Ziliang Chen, Guangrun Wang, Xiaodan Liang, and Liang Lin. Law-diffusion: Complex scene generation by diffusion with layouts, 2023a. URL https://arxiv.org/abs/2308.06713.

Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and Bin CUI. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal LLMs. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=DgLFkAPwuZ.

Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Idea2img: Iterative self-refinement with gpt-4v(ision) for automatic image design and generation. *arXiv preprint arXiv:2310.08541*, 2023b.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

Lirui Zhao, Yue Yang, Kaipeng Zhang, Wenqi Shao, Yuxin Zhang, Yu Qiao, Ping Luo, and Rongrong Ji. Diffagent: Fast and accurate text-to-image api selection with large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6390–6399, June 2024.

Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22490–22499, June 2023.

A TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

CONTENTS

A.1 Discussions about StructMap 17
A.2 Detailed Experiment Settings 18
A.2.1 Dataset Construction 18
A.2.2 Implementation Details 18
A.2.3 Baseline Methods 18
A.2.4 Discussion on Evaluation Metrics 19
A.2.5 Human Evaluation 19
A.3 More Qualitative Results 20
A.4 Viewpoint and Structural Editing 23
A.5 Ablation Studies 23
A.5.1 Generality of SAGA under Different MLLMs 23
A.5.2 the Effectiveness of StructMap 24
A.5.3 The Effectiveness of SAGA 26
A.5.4 The Effectiveness of the Realism Memory Module 27
A.6 Comparison with 3D generation methods 28
A.7 Direct Evaluation of Structural Fidelity 29
A.8 Higher-order Configurations in StructMap Design Interface 30
A.9 Detailed Prompts and Responses in SAGA 31
A.10 the Complexity of StructMaps in Our Dataset 36
A.11 Number of Iterations in SAGA, API Response Time and Failure Cases 37
A.12 Generating Other Categories of Images 37
A.13 Limitation 38

A.1 Discussions about StructMap

Motivation and Rationale Behind StructMap Our goal is to develop a 3D representation that can flexibly and accurately encode an object's topology and spatial layout, while remaining intuitive and accessible for users to create and revise. The most straightforward approach is to characterize an object's structure using points and lines Haridis (2020); Xu et al. (2025), where each line represents a component of the object and each point represents the connections between components, as illustrated in Fig. 8. However, since points and lines have no 3D volume, this representation cannot capture the specific size and shape of each component. Such natural disadvantage makes it difficult to represent certain complex topology and it also prevents precise specification of the connection locations between components, such as a seat with four legs attached precisely at the quarter points along its diagonals. To overcome these limitations, we propose representing each basic component as a 3D geometric primitive with controllable size and pose, thereby enabling a finer-grained representation of their spatial relationships and connections, such as the example illustrated in Fig. 8, two legs can be attached to the seat at any position and in any manner. Building on our observations of a wide range of real-world objects, we carefully select a set of representative geometric primitives and design shape-controlling parameters for them. This ensures that the primitives can effectively encode the topology and spatial layout of diverse objects while remaining simple and convenient for users to manipulate. Moreover, this design also ensures the extensibility of the representation, as additional primitives and richer parameterizations can be introduced to support scalable modeling across various object categories. In addition, this representation offers a cognition-inspired approach to model object structure, aligning with how humans perceive and reason about 3D objects. According to classic theories in cognitive science, particularly the 'Recognition-by-Components' theory Biederman (1987), humans recognize objects not by memorizing textures and geometric details, but by decomposing them into a set of simple, geometric primitives along with their spatial configuration.

StructMap vs. Sketch While both StructMap and sketches are able to describe an object's structure, they differ in dimensionality and expressiveness, which leads to different performance in structure-grounded object image generation. (1) Sketches are inherently 2D and manually drawn, requiring high professional skills yet still being difficult to faithfully capture 3D structural properties such as depth, angle of view, and symmetry. In contrast, StructMap is natively 3D, and thus naturally encodes such 3D attributes. Moreover, sketches struggle to maintain viewpoint and structural consistency; that is, creating coherent sketches of the same object from different viewpoints or with modified structural configurations is nontrivial, as each new view or structural variation requires redrawing and careful recalibration. In comparison, once a StructMap is constructed, rendering from different viewpoints or applying structural edits (e.g., modifying part configurations or spatial arrangements) becomes trivial and consistent, without requiring additional effort. This property naturally supports appearance-consistent viewpoint editing and structural editing in our framework. (2) StructMap offers a more accessible structural input condition. Unlike sketches that demand drawing skills, StructMap creation only requires assembling geometric primitives via interpretable parameters. This significantly reduces the creation burden and skill barrier for users. Alternatively, attempts to simplify sketching, like using scribbles, often result in significant information loss, leading to incomplete or ambiguous structural specification.



Figure 8: **[top]** An example of how to represent the structure of a chair using points and lines. **[bottom]** An example illustrating two different ways in which two legs are connected to a seat. This example demonstrates that our representation enables a finer-grained characterization of the spatial relationships and connection between components.

Structure Modeling System vs. CAD Unlike traditional Computer-Aided Design (CAD) tools, which are primarily geared towards precise and detailed 3D modeling typically used in engineering and industrial design, our structure modeling system focuses on simplicity, accessibility, and task-specific functionality. CAD systems often require significant professional expertise, extensive training, and a deep understanding of geometrical constraints(e.g., parallelism, symmetry, perpendicularity, concentricity). They are powerful, but their complexity can be a barrier for non-expert users or for applications where such precision is unnecessary. In contrast, our structure modeling system is

designed specifically for structural representation rather than comprehensive 3D object modeling. It provides an intuitive, streamlined interface that allows users—even those without a background in engineering or design—to quickly create, manipulate, and visualize structural models. By prioritizing usability and focusing on the essential features needed for structure modeling, our system significantly lowers the entry barrier, enabling easy use by anyone.

A.2 Detailed Experiment Settings

A.2.1 Dataset Construction

To construct our structure-prior dataset, we first invite human creators to build a diverse set of StructMaps that span a wide range of object categories, topologies, and spatial layouts using our StructMap design system. Based on each designed StructMap, the creators are then instructed to manually create corresponding textual descriptions, linearts, and scribbles that depict the same underlying structural configuration. When drawing linearts and scribbles, the creators are free to choose a suitable viewpoint that they feel best conveys the structure. This ensures that each set of inputs — text, lineart, scribble, and StructMap — corresponds to the same structural intent.

For the linearts, creators are encouraged to preserve as much structural detail as possible, closely match the reference StructMap, and apply light shading where appropriate to enhance three-dimensionality. However, in practice, we found that most creators, due to varying levels of technical skill, often produce line drawings with errors and noise. Since our dataset aims to provide high-quality lineart, we leverage MLLM-based optimization techniques to enhance the line drawings, thereby reducing the technical demands and time investment required from creators, while still ensuring that the resulting lineart meets the demands of our dataset. Scribbles, in contrast, are designed to emphasize simplicity and accessibility: creators only need to roughly retain the main structure of the object, and we do not impose strict quality requirements.

It is important to note that our dataset construction does not require specific consideration of artistic style, as our primary focus is on the structural information within the data. Artistic style often influences the fine-grained appearance details of the objects, which are not the main concern of this dataset.

To help readers better understand the acquisition process and difficulty level of each type of input condition, we recorded a video showcasing the manual steps involved in creating all four types of inputs side by side. This can be found in the supplementary video file *all_conditions_creation.mp4*.

A.2.2 Implementation Details

We use *Open3D* to render the 3D StructMap into 2D images with default settings, including Lambertian shading, neutral directional lighting, and uniform gray mesh color. The rendered images have a resolution of 512×512 pixels and are captured from carefully selected viewpoints to ensure key structural features are clearly visible. SAGA based on GPT Image 2 is executed via the official OpenAI API. Full prompts used in SAGA are provided in Section A.9.

A.2.3 Baseline Methods

To comprehensively evaluate our method, we compare against strong baselines under each conditioning input. For each condition, we select two representative models that are either widely adopted or represent state-of-the-art advancements in that condition. Here, we will provide a detailed description of each method we have selected.

- **Text-based Condition.** We compare against **OmniGen** Xiao et al. (2024), a unified image generation model that excels in multiple domains. OmniGen demonstrates competitive text-to-image generation capabilities and inherently supports a variety of downstream tasks, such as controllable image generation and classic computer vision tasks. We also compare against **Lumina-Image 2.0** Qin et al. (2025), a unified and efficient T2I generative framework. Lumina-Image 2.0 adopts a unified architecture (Unified Next-DiT) that treats text and image tokens as a joint sequence, allowing for seamless cross-modal interaction and enabling task expansion without retraining, leading to improved image fidelity and alignment with text inputs.

- **Lineart-based Condition.** We compare against **ControlNet++** Li et al. (2024), an enhanced version of ControlNet that improves controllability by leveraging efficient consistency feedback mechanisms. Additionally, we evaluate **ControlAR** Li et al. (2025), an advanced framework that integrates spatial controls into autoregressive image generation models. It generates the next image token based on control and image tokens' fusion, similar to positional encoding, thus enhancing the autoregressive model's control capabilities.
- **Scribble-based Condition.** We compare against **ControlNet** Zhang et al. (2023), a widely used framework for guided image generation from edge and sketch inputs. ControlNet enables the generation of high-quality images based on simple sketches or outlines, allowing for precise control over the structure and composition of the generated visuals. Another comparison is with **T2I-Adapter** Mou et al. (2023), a lightweight plug-and-play adapter that facilitates structure guidance for existing diffusion models. T2I-Adapter achieves structural control over the generation process by learning the alignment between internal knowledge of text-to-image models and external control signals.

A.2.4 Discussion on Evaluation Metrics

In our work, we report common metrics such as SSIM and LPIPS, which are widely used in prior works for evaluating image similarity. However, these metrics mainly measure pixel-level or feature-level similarity and are not specifically designed to assess the alignment of topology and spatial layout. Our method focuses on preserving structural properties such as topology and spatial layout, which go beyond pixel or feature accuracy. Therefore, while SSIM and LPIPS provide useful references, they may not fully capture the structural alignment that our method aims to achieve. To complement these metrics, we additionally rely on human evaluation to better assess topology and spatial layout consistency. Developing more suitable automatic metrics for this purpose remains an important direction for future work.

A.2.5 Human Evaluation

We conduct a two-part Mean Opinion Score (MOS) study Huynh-Thu et al. (2011), where human participants evaluate (1) the realism of generated images (**MOS-R**) and (2) the alignment between the generated image and the input structure (**MOS-A**) on a 1–5 Likert scale Likert (1932). Fig. 9 shows an example of our scoring system.

The detailed scoring criteria are defined as follows:

MOS-R (Realism):

- **1 – Very Poor:** The image is clearly artificial and unrealistic.
- **2 – Poor:** The image contains many unrealistic artifacts and lacks visual plausibility.
- **3 – Fair:** The image is somewhat realistic but contains noticeable flaws or inconsistencies.
- **4 – Good:** The image appears mostly realistic with only minor issues.
- **5 – Excellent:** The image is highly realistic and could plausibly be mistaken for a real photograph.

MOS-A (Alignment with Structure):

- **1 – Very Poor:** The image does not reflect the input structure at all.
- **2 – Poor:** Only a few structural elements are preserved; major components are missing or distorted.
- **3 – Fair:** The overall layout is partially aligned with the input, but several elements are misrepresented.
- **4 – Good:** The image largely follows the structural configuration, with only minor deviations.
- **5 – Excellent:** The image shows an almost perfect correspondence with the input structure.

Specifically, we collected 40 human evaluation responses, including 20 participants with experience in computer vision or graphics and 20 participants without relevant domain expertise. Each participant

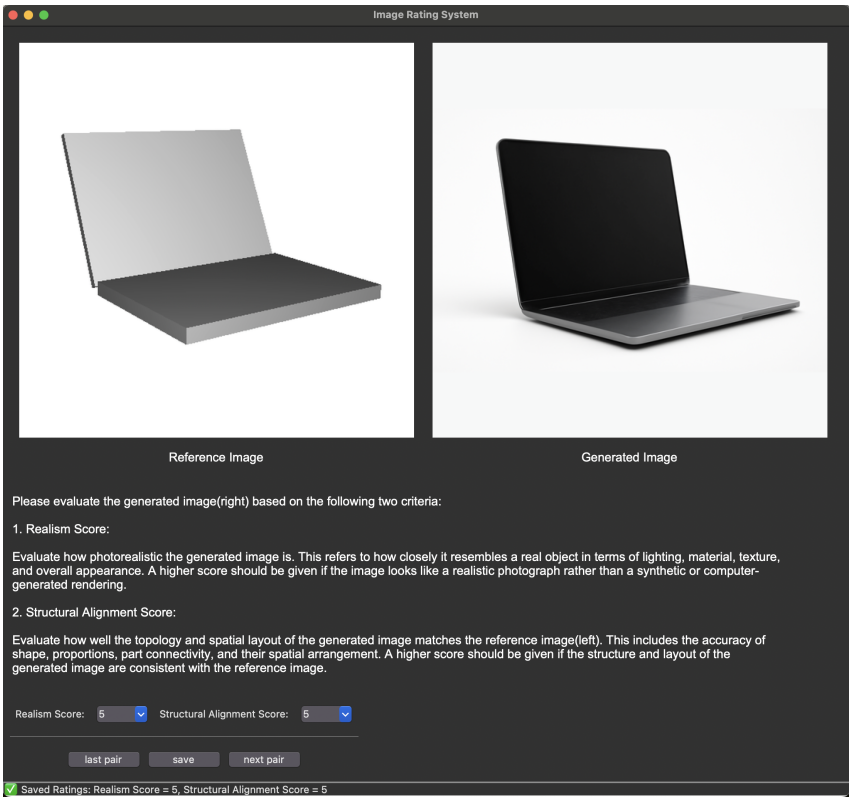

Figure 9: An example of the scoring system.

was asked to rate a randomly sampled subset of generated images using the scoring criteria described above. To reduce potential bias, the order of samples was randomized for each participant, and participants were not informed of the underlying generation method used to produce the images. Each image was evaluated by multiple participants, and the final MOS-R and MOS-A scores were computed by averaging the ratings across all collected responses. Participants were compensated at a rate of 20 USD per hour.

A.3 More Qualitative Results

To further demonstrate the generality and effectiveness of Struct2Real across diverse object categories, we provide additional qualitative comparisons in Fig. 10 and Fig. 11, and further analyze the performance based on these results.

Realism of the Generated Images. Our qualitative results further confirm that our method consistently generates object images with higher visual realism. In many cases, the generated results appear convincingly realistic, with photorealistic textures, natural shading, and plausible materials. By contrast, baseline methods often display visual artifacts such as distorted geometry(Scribble & Controlnet in example (c)) or artificial textures(Lineart & Controlnet++ in example (g)).

Structural Alignment with the Conditioning Input. From visual comparisons, we find that text- and scribble-conditioned outputs often exhibit incorrect structural attributes, such as misplaced components(Scribble & Controlnet in example (i)) or missing parts(Text & Omnigen in example (f)), due to the ambiguity or incompleteness of the input. Lineart provides much stronger constraints, but may inadvertently introduce noise(Lineart & Controlnet++ in example (n)) or overfitting to edges(Lineart & controlar in example (d)), which can hinder natural appearance. In contrast, our method reliably maintains key structural properties, even if minor variations in orientation or scale occasionally occur.

Additionally, due to space limitations in the figure, we do not show the full text conditions. Below, we provide the complete text conditions for (a)–(d) as illustrative examples.



Figure 10: More qualitative results across different structural conditions and approaches. There are example (a)-(g). For each sub-panel, the top row shows different types of structural conditions, and the bottom row shows the corresponding generated object images. The text condition in the figure appears small and may require zooming in for a clear view.



Figure 11: More qualitative results across different structural conditions and approaches. There are example (h)-(n). For each sub-panel, the top row shows different types of structural conditions, and the bottom row shows the corresponding generated object images. The text condition in the figure appears small and may require zooming in for a clear view.

(a) Create an image of a cooking pot consisting of a simple wide cylindrical body with straight vertical walls, featuring two blocky rectangular side handles extending outwards from the upper exterior sides, each formed by two parallel struts connected by a flat crossbar, and a central handle structure positioned centrally above the opening comprised of two vertical rectangular posts supporting a horizontal flat rectangular bar.

(b) Create an image of a desk consisting of a flat rectangular tabletop supported by a frame with inverted T-shaped legs made of square-profile beams, featuring a suspended cubic drawer unit on the right side with three vertically stacked drawers each having a horizontal bar handle, a vertical rectangular modesty panel at the back connecting the left leg to the drawer block, and a rectangular side panel attached to the lower inner section of the left leg frame.

(c) Create an image of a mug characterized by a main cylindrical body with straight vertical walls and a visible thickness at the flat upper rim, featuring a deep hollow interior and a smooth, C-shaped tubular handle attached to the side that curves outward to connect the upper and lower sections of the cylinder.

(d) Create an image of a cubic safe, featuring a rectangular door panel defined by a recessed outline on the front face, articulated by two protruding cylindrical hinges along the right vertical edge, with two vertically aligned circular depressions in the center of the door where the top one forms a ring-like dial housing and the bottom one contains a geometric three-pronged handle shape, viewed from an angle that reveals the flat left side and the detailed front facade.

A.4 Viewpoint and Structural Editing

In this section, we provide additional qualitative examples to further demonstrate the effectiveness of Struct2Real in viewpoint and structural editing. As shown in Fig. 12, Struct2Real consistently produces images that faithfully follow the intended viewpoint and structural edits while maintaining consistent geometric details and texture-material attributes with the original images. These results stem from two key factors: (i) StructMap provides an explicit and accurate representation of viewpoint and structural modifications, and (ii) the design of SAGA, including its skill-based generation mechanism and realism memory module, ensures consistency in both structure and appearance during editing. In contrast, generations based on alternative input conditions with MLLMs frequently exhibit noticeable artifacts, including misaligned viewpoints (Lineart & GPT in example (c)-[Left]), structural inconsistencies (Text & GPT in example (a)-[Left]), and appearance discrepancies (Scribble & GPT in example (d)-[Right]).

A.5 Ablation Studies

To further analyze the effectiveness of Struct2Real in generating photorealistic object images under topology and spatial layout constraints, we conduct a series of ablation studies. These studies are designed to address the following questions: 1) How generalizable is SAGA across different underlying MLLMs? 2) What is the contribution of StructMap compared to other condition types (e.g., text, scribble and lineart)? 3) How effective is our proposed SAGA?

A.5.1 Generality of SAGA under Different MLLMs

To assess the generality of SAGA, we replace the underlying MLLM of the components with alternative models. Specifically, we substitute the MLLM with four alternative models: Gemini 3 Pro Pichai et al. (2025), Qwen-image-edit Wu et al. (2025), Flux-1-kontext-dev Labs et al. (2025), and Seedream-4 Gao et al. (2025). We first compare the performance of SAGA under different MLLMs against baseline methods, and then conduct comparisons across different MLLMs. Related results are presented in Fig. 13.

From Fig. 13, we observe that baseline methods (right) often produce results with limited visual realism (e.g., Text & Lumina result in example (c)) or noticeable structural inconsistencies (e.g.,



Figure 12: Qualitative results of the viewpoint and structural editing experiment. For each sub-panel, the top row shows the input structural conditions, while the bottom row shows the corresponding generated object images. The text condition in the figure appears small and may require zooming in for a clear view.

Scribble and Controlnet result in example (e)). In contrast, SAGA consistently yields more realistic and structurally faithful results across all tested MLLMs. Among them, results generated using GPT Image 2 as the underlying model exhibit the highest visual realism (e.g., StructMap & GPT Image 2 result in example (e)).

A.5.2 THE EFFECTIVENESS OF STRUCTMAP

To evaluate the effectiveness of StructMap, we replace the input to SAGA with alternative conditions: text, lineart, and scribble. These conditioning inputs are obtained from our structure-prior dataset, where all conditions are aligned to the same object structure. To ensure fairness and correctness, we slightly adjust the prompting strategy of SAGA for different input types. We compare the performance of StructMap with the three alternative conditions. Related results are shown in Fig. 14.

From Fig. 14, we observe that while all four conditioning inputs produce outputs of similar visual realism, their structural consistency varies significantly. Specifically, results conditioned on text and scribble exhibit poor alignment with the intended structures (e.g. Text result in example (c) and Scribble result in example (d)). This indicates that such conditions provide insufficient structural information, limiting SAGA's ability to generate structurally faithful images. In contrast, conditioning on our proposed StructMap produces outputs that reliably preserve the underlying topology and spatial layout, highlighting its effectiveness in providing accurate structural guidance. While results conditioned on lineart achieve a comparable level of structural consistency, StructMap offers clear advantages in terms of acquisition efficiency and ease of creation, making it a more practical and scalable option for structure-grounded generation.



Figure 13: Qualitative comparison of generated images when the underlying model of SAGA is substituted with four alternative MLLMs. Generated outputs are compared against baseline methods. For each sub-panel, the top row shows the input structural conditions, while the bottom row shows the corresponding generated object images. The text condition in the figure appears small and may require zooming in for a clear view.

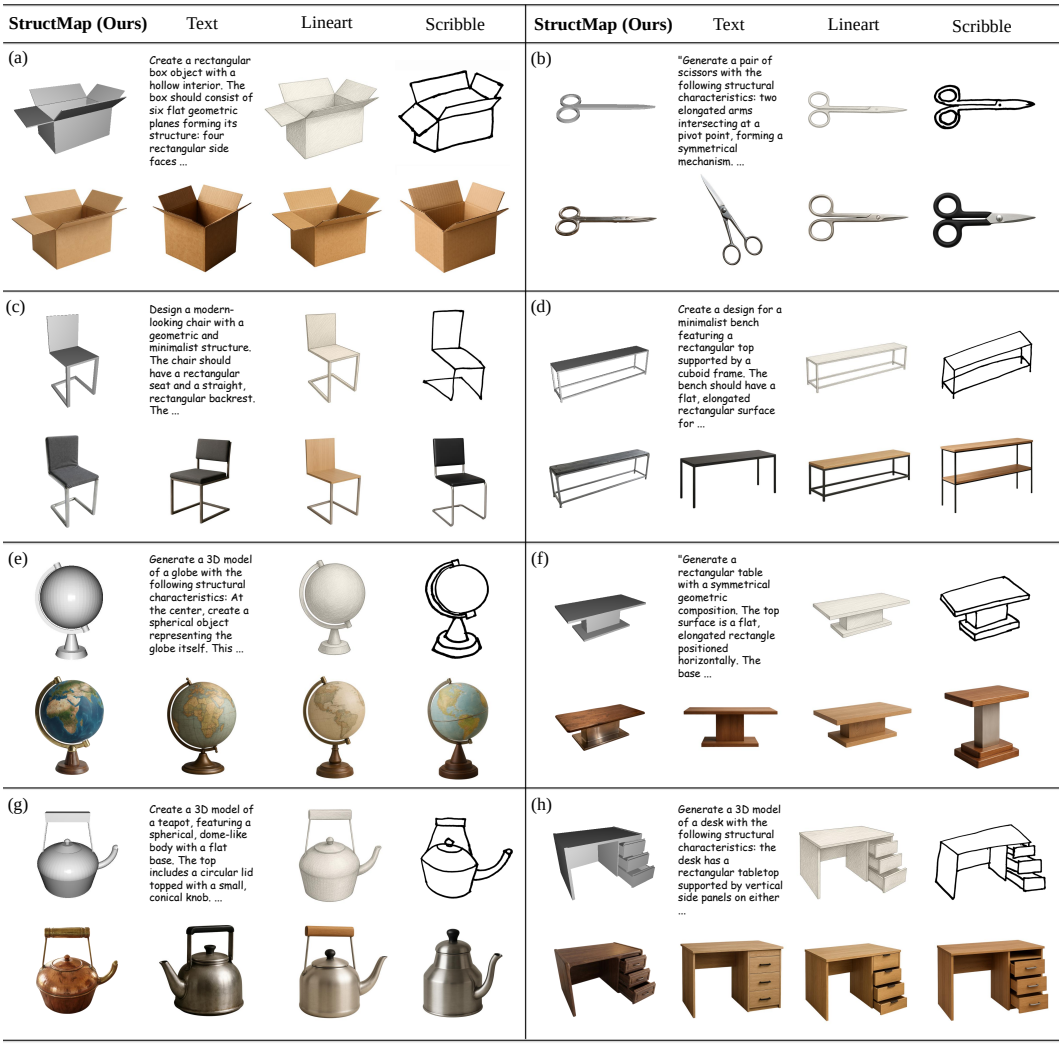

Figure 14: Qualitative comparison of image generation results under different conditioning inputs. For each sub-panel, the top row shows different types of structural conditions, and the bottom row shows the corresponding generated object images. The text condition in the figure appears small and may require zooming in for a clear view.

A.5.3 The Effectiveness of SAGA

To evaluate the effectiveness of our proposed SAGA, we first compare its outputs with those generated by directly employing a single MLLM, with both approaches conditioned on the same StructMap. Qualitative results are presented in Fig. 15. As shown, directly using an MLLM often leads to structural inconsistencies (e.g., the w/o SAGA results in examples (a) and (b)) or reduced visual realism (e.g., the w/o SAGA results in examples (d) and (f)). In contrast, SAGA consistently produces images that are both visually realistic and structurally faithful, demonstrating the benefit of the skill-oriented iterative generation process.

To further investigate the contribution of individual skills in SAGA, we remove the three skills one at a time and perform image generation under each configuration. An illustrative example is presented in Fig. 16. We observe that the Structure Preservation Skill effectively corrects structural inconsistencies, the Detail Enhancement Skill introduces richer and more realistic geometric details, and the Appearance Refinement Skill improves texture and material realism. These qualitative observations suggest that the three skills play complementary roles in achieving high-quality structure-grounded image generation.

We further conduct a quantitative skill-level ablation on a stratified subset of 400 StructMaps covering 20 object categories and different levels of structural complexity. We report SSIM and LPIPS for image-level structural similarity, together with MOS-A and MOS-R for human evaluation of structural alignment and visual realism, respectively. We omit FID in this experiment because its distribution-level estimate is less reliable on this relatively small subset. The results are summarized in Tab. 2.

As shown in Tab. 2, Full SAGA achieves the best performance across all evaluation metrics. Removing the Structure Preservation Skill causes the largest decrease in MOS-A, from 4.74 to 4.03, confirming its importance for maintaining structural fidelity. In comparison, removing the Detail Enhancement Skill or Appearance Refinement Skill leads to more pronounced degradation in MOS-R, reflecting their respective roles in enriching geometric details and improving texture and material realism. Furthermore, removing the entire SAGA framework results in an additional performance decrease across all metrics. These results quantitatively demonstrate the complementary contributions of the three specialized skills and the benefit of coordinating them within the SAGA framework.

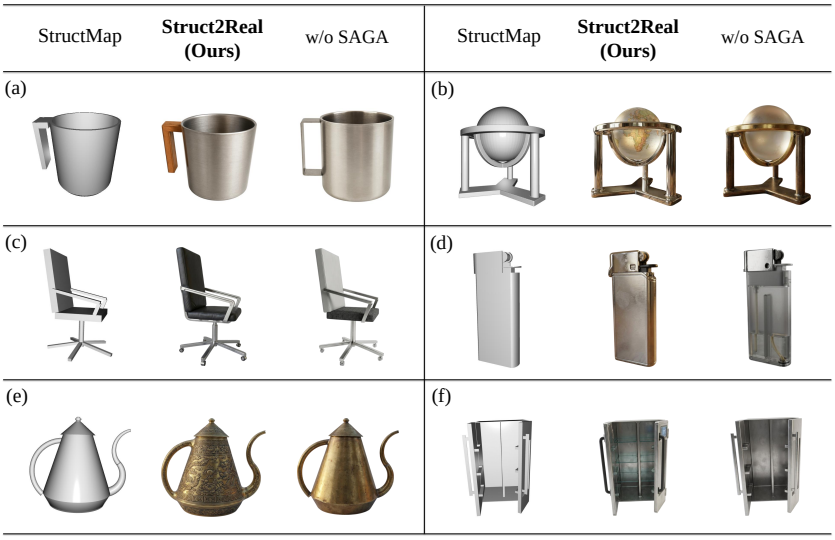

Figure 15: Qualitative comparison between SAGA and directly employing a single MLLM, both conditioned on StructMap.

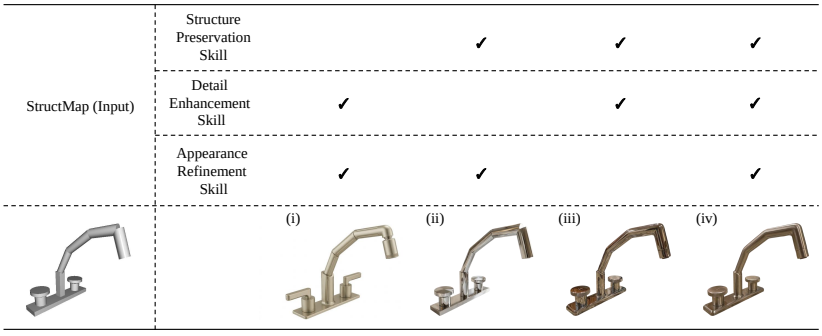

Figure 16: Illustrative example of the ablation study on different component configurations in SAGA.

A.5.4 The Effectiveness of the Realism Memory Module

The Realism Memory Module is designed to preserve previously established visual attributes when users perform viewpoint or structural editing. To evaluate its contribution, we conduct an additional ablation study on 30 editing cases involving both viewpoint changes and structural modifications. For each case, the two configurations receive exactly the same initial generated image, modified StructMap, and editing instruction. The only difference is whether the part-level visual attributes recorded during the previous generation process are reused during editing.

Table 2: Quantitative ablation of different SAGA configurations on a stratified subset of 400 StructMaps covering 20 object categories and different levels of structural complexity.

Configuration	SSIM↑	LPIPS↓	MOS-A↑	MOS-R↑
w/o SAGA	0.642	0.134	3.86	4.02
w/o Structure Preservation	0.659	0.121	4.03	4.36
w/o Detail Enhancement	0.686	0.109	4.48	4.18
w/o Appearance Refinement	0.695	0.106	4.57	4.09
Full SAGA	**0.704**	**0.091**	**4.74**	**4.77**

We evaluate the edited results along four dimensions: Edit Compliance, Structural Fidelity, Appearance Consistency, and Visual Realism, all on a 1–5 scale. Edit Compliance measures whether the requested viewpoint or structural modification is correctly performed; Structural Fidelity evaluates whether the edited result follows the modified StructMap; Appearance Consistency measures how well the visual attributes of unchanged parts are preserved with respect to the original generated image; and Visual Realism evaluates the overall realism of the edited image. The quantitative results are summarized in Tab. 3.

Table 3: Quantitative ablation of the Realism Memory Module on 30 viewpoint and structural editing cases. All metrics are evaluated on a 1–5 scale.

Editing Configuration	Edit Compliance↑	Structural Fidelity↑	Appearance Consistency↑	Visual Realism↑
w/o Realism Memory	4.56	4.51	3.84	4.38
Full SAGA	**4.62**	**4.68**	**4.62**	**4.70**

As shown in Tab. 3, both configurations achieve similarly high Edit Compliance scores, indicating that the requested viewpoint or structural modifications can generally be executed even without the Realism Memory Module. However, removing the memory causes a substantial decrease in Appearance Consistency, from 4.62 to 3.84. This suggests that, without explicitly reusing previously recorded visual attributes, the editing process is more likely to introduce unintended changes to the appearance of parts that are not targeted by the edit.

We also observe moderate decreases in Structural Fidelity and Visual Realism when the memory is removed. One possible explanation is that the stored visual attributes provide additional context about the previously established geometry and appearance, helping the model maintain coherent part-level details while adapting the object to the modified structural condition. Overall, these results indicate that the primary role of the Realism Memory Module is not simply to enable the requested edit, but to preserve previously established geometric details, material properties, texture characteristics, and overall visual identity throughout the editing process.

A.6 Comparison with 3D generation methods

We observed that, beyond conventional image generation methods, one can also leverage 3D generation methods by first creating 3D objects and then rendering them into images. So in this section, we evaluate the performance of 3D generation methods on our proposed task of structure-aware image generation. Specifically, we use four types of conditions from our structure-prior dataset, apply state-of-the-art text-to-3D or image-to-3D methods to generate 3D meshes, and then render images from these meshes to obtain realistic object images. In practice, we employ Hunyuan3D Team (2025) as the text-to-3D and image-to-3D model in our experiments.

As shown in Fig. 17, the performance of these 3D generation methods is generally unsatisfactory. When using text and scribble as conditions, the results exhibit very poor structural fidelity, since text and scribble cannot accurately and completely convey structural information. By contrast, when conditioned on lineart and StructMap, the generated results achieve better structural fidelity but suffer from low realism. This is because image-to-3D methods are designed to faithfully reconstruct the objects present in the input images. Consequently, imperfections in lineart drawings and discontinuous connections between geometry primitives in StructMap are also preserved. Moreover, the textures of

the generated meshes are determined by those in the input images. As our conditional images do not provide any texture information, the resulting meshes inevitably contain unrealistic textures.

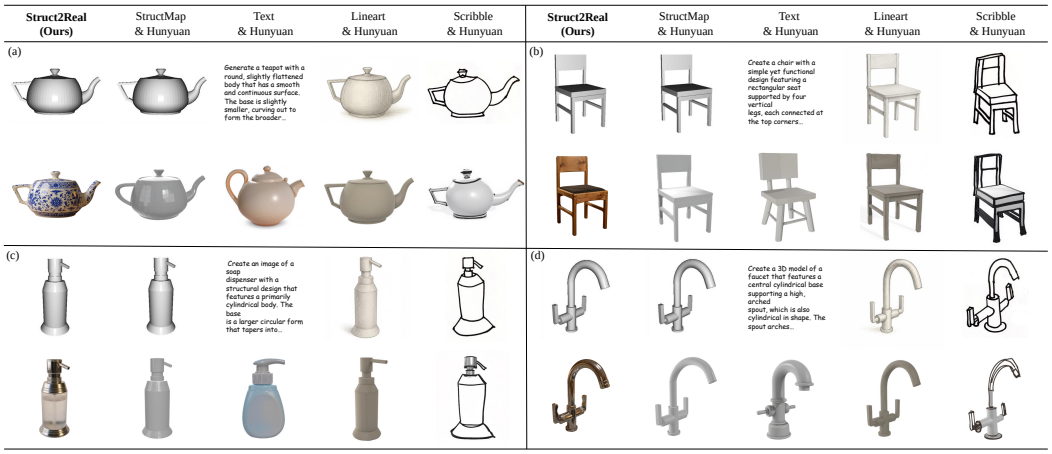

Figure 17: Qualitative results across different 3d generation methods. For each sub-panel, the top row shows the input conditions for each method, and the bottom row displays the corresponding generated object images. The text condition in the figure appears small and may require zooming in for a clear view.

A.7 Direct Evaluation of Structural Fidelity

Our main evaluation reports SSIM, LPIPS, and MOS-A to measure the alignment between generated images and structural conditions. However, SSIM and LPIPS primarily capture pixel-level or feature-level similarity, while MOS-A provides an overall judgment of structural alignment. To more directly evaluate the structural properties emphasized in our task, we conduct an additional focused evaluation along five dimensions: *part count*, *connectivity*, *relative position*, *relative pose*, and *viewpoint*. These dimensions directly characterize the topology and spatial layout that StructMap is designed to encode. Each structural dimension is evaluated on a 1–5 scale, where a higher score indicates stronger agreement between the generated image and the input structural condition.

MLLM-based evaluation. We first employ GPT-5.6 Sol as an independent evaluator. For each sample, the evaluator receives the rendered StructMap, a structural evaluation checklist, and an anonymized generated result. Method identities are hidden and the presentation order is randomized to reduce evaluation bias. Each sample is evaluated three times independently, and the median score is reported. The results are summarized in Tab. 4.

Table 4: Direct structural fidelity evaluation using an independent MLLM evaluator. All dimensions are scored on a 1–5 scale.

Method	Part Count↑	Connectivity↑	Rel. Pos.↑	Rel. Pose↑	Viewpoint↑	Avg.↑
Text + GPT Image 2	2.65	2.05	2.28	1.98	2.50	2.29
Scribble + GPT Image 2	4.05	3.78	3.91	3.62	3.86	3.84
Lineart + ControlNet++	4.67	4.48	4.57	4.39	4.64	4.55
Struct2Real	**4.83**	**4.79**	**4.81**	**4.73**	**4.80**	**4.79**

Human evaluation. We also conduct a blinded human study using the same five-dimensional rubric. Ten participants take part in the evaluation. Each generated result is independently scored by at least three participants, with method identities hidden and sample order randomized. The final score for each dimension is obtained by averaging all collected ratings. The results are reported in Tab. 5.

Both the MLLM-based and human evaluations exhibit highly consistent trends. Struct2Real achieves the highest score across all five structural dimensions, indicating stronger preservation of part

Table 5: Direct structural fidelity evaluation by human participants using the same five-dimensional rubric. All dimensions are scored on a 1–5 scale.

Method	Part Count↑	Connectivity↑	Rel. Pos.↑	Rel. Pose↑	Viewpoint↑	Avg.↑
Text + GPT Image 2	2.58	1.99	2.22	1.93	2.42	2.23
Scribble + GPT Image 2	3.98	3.71	3.85	3.56	3.79	3.78
Lineart + ControlNet++	4.63	4.44	4.53	4.34	4.60	4.51
Struct2Real	**4.79**	**4.74**	**4.76**	**4.69**	**4.76**	**4.75**

composition, connectivity, spatial relationships, relative pose, and viewpoint. The improvement is particularly notable in connectivity, relative position, and relative pose, which are difficult to capture using general image-similarity metrics alone. These results provide more direct evidence that StructMap enables accurate control over object topology and spatial layout, complementing the SSIM, LPIPS, and MOS-A results reported in the main evaluation.

A.8 Higher-order Configurations in StructMap Design Interface

In addition to primitive-level composition, our design interface supports a collection of higher-order configurations distilled from frequently occurring structural patterns in real-world objects. We impose certain constraints on the degrees of freedom of the primitives within these higher-order configurations and design corresponding parameters to control the adjustable attributes of the primitives in these configurations. This allows users to instantiate a higher-order configuration to create multiple primitives simultaneously, enabling more primitives to be created with fewer parameters and thereby improving creation efficiency. For example, in symmetric structures, we can use a single parameter to jointly control the parameters of the two primitives on both sides of the symmetry axis. We have designed a total of 40 higher-order configurations, and representative examples of the higher-order configurations are visualized in Figure 18.

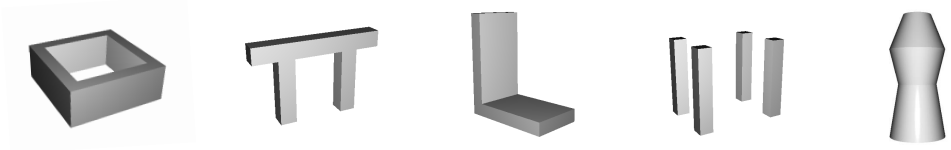

Figure 18: Representative examples of the higher-order configurations.

A.9 Detailed Prompts and Responses in SAGA

Initialization Prompt. The initialization prompt defines the fundamental task specification for the SAGA agent at the beginning of the generation process. It clarifies the semantic meaning of the StructMap representation and specifies the objective of the synthesis task.

> ***StructMap Properties:***
> *This image is a StructMap of an object. It is composed of basic geometric primitives such as cuboids, cylinders, toruses, and spheres. The StructMap represents the complete topology, geometric composition, and relative size proportions of the object. It describes the structural skeleton of the object but does not contain texture, material, or lighting information.*
> ***Generation Objective:***
> *Based on this structural skeleton, generate a realistic image of the corresponding real-world object:*
> *1. Strictly preserve the topology, geometric composition, and relative size proportions defined by the StructMap.*
> *2. The generated object should resemble a realistic real-world object. Introduce slight natural deformations and geometric details while maintaining smooth connections between primitives.*
> *3. Add realistic textures, materials, and lighting that are consistent with the object's physical properties.*
> *4. Preserve exactly the same camera viewpoint as the StructMap image.*

Orchestration Core Prompt. The orchestration core prompt is used after the initialization stage, where the StructMap image and the currently generated image are jointly provided to the agent. Its purpose is to evaluate the current generation result from the perspectives of structural consistency and visual realism, and then determine the next refinement action.

> *You are a professional expert in 3D object generation evaluation. Now you are given two images:*
> *- The first image: the original structural StructMap image (reference).*
> *- The second image: the currently generated realistic object image (to be evaluated).*
> *Please compare the two images and strictly evaluate the generated result. Return the output in JSON format only (do not add "'json or any extra content).*
> *Evaluation dimensions:*
> *1. Structural aspect: whether the topology, geometric composition, size proportions, and viewpoint are fully consistent with the structural StructMap.*
> *2. Detail aspect: whether reasonable geometric details have been added, such as bevels, smooth transitions, and natural connections without abrupt sharp edges.*
> *3. Appearance aspect: whether the texture, material, and lighting are natural, realistic, and visually plausible.*
> *Available actions:*
> *- structure_preservation: the structure, proportions, or viewpoint are inconsistent with the StructMap and need to be corrected.*
> *- detail_enhancement: the geometric details are insufficient, such as missing bevels, transitions, or natural connections.*
> *- appearance_refinement: the material, texture, or lighting is unrealistic or visually unsatisfactory.*
> *- exit: the structure is correct, the geometric details are sufficient, and the appearance is realistic and satisfactory.*
> *Return format:*
> *{"action": "xxx", "reason": "a brief explanation of the identified issue"}*

Structural Description Prompt (in Structure Preservation Skill). This prompt is used to generate a textual structural description of the StructMap when structural inconsistencies are detected during generation.

You are a 3D structural analysis assistant.
You are given several structural conditions derived from a StructMap of an object, including:
- the StructMap image itself
- multi-view renderings of the StructMap
- the normal map of the StructMap
- the position map of the StructMap
These inputs together describe the structure of the object from multiple perspectives.
Your task is to produce a clear textual description of the object's structure.
The description should include:
1. the number of structural components
2. the type of each component (e.g., cuboid, cylinder, torus, sphere)
3. the spatial arrangement of the components
4. the relative size relationships
5. the connectivity and attachment relationships between components
Focus only on topology and spatial layout.
Do not describe texture, material, color, or lighting.
The description should be concise but sufficiently detailed to fully convey the structural organization of the object.

Textual Structural Description [Model Response] (in Structure Preservation Skill). This is MLLM's response in structure preservation skill.

Here is a detailed 3D structural analysis of the chair-like object, including its geometric primitives, spatial relationships, connections, and viewing orientation:

1. Primitive Shapes and Detailed Features The object is composed entirely of cuboids (rectangular prisms). Here's a breakdown:
a. Backrest
Shape: Vertically oriented cuboid
Size & Proportions: Wide and flat (stretched along the horizontal and vertical axes, thin along the depth)
Orientation: Upright; perpendicular to the seat
Position: Attached to the rear of the seat, elevated above it
b. Seat
Shape: Flat cuboid
Size & Proportions: Square top surface, shallow height
Color: Darker than the rest of the structure (possibly meant to indicate cushion material)
Orientation: Horizontally flat, aligned with the base plane
Position: Central structural element, connects all leg elements and backrest
c. Legs (4)
Shape: Uniform cuboids (not tapered)
Size & Proportions: Tall and narrow (long vertical axis, short cross-section)
Orientation: Vertically aligned, each positioned near a seat corner
Position: Extend downward from the seat to the floor
d. Support Bars (3 visible)
Shape: Thin cuboids
Size & Proportions: Thin and narrow; shorter than legs
Orientation: Horizontally aligned
Position: Two connect the front and back legs (left and right side), one connects the side legs at the back
Function: Likely structural reinforcement

2. Relative Sizes and Spatial Relationships
Backrest is vertically the tallest single component and thinner than the seat.
Seat is the largest surface area component, proportionally wider than it is tall.

Legs are all of equal size and symmetrically placed under the seat.
Support bars are thinner and shorter than legs but positioned at mid-height between the seat and floor, maintaining the structural frame.

3. Connectivity and Alignment
Backrest is directly attached to the rear of the seat.
Legs are flush-mounted to the four corners of the seat underside — no overhang.
Support bars are aligned with leg centers and intersect or touch the inner faces of the legs.
All joints appear cleanly orthogonal — no visible rotation, tapering, or angular misalignment.

4. Viewing Angle and Orientation
The object is rendered from a slightly elevated isometric view — a 3D angle showing:
Front-right perspective (both front and right legs are visible)
Top surface of the seat is also partially visible
The backrest is facing away from the viewer
Camera is positioned at an angle slightly above and to the right of the object's center, creating a 3-point perspective-like view

Regeneration Prompt (in Structure Preservation Skill). After providing additional structural conditions, including multi-view renderings, normal maps, position maps, and the textual structural description of the StructMap, the agent is instructed to regenerate the image while enforcing stronger structural consistency.

You are refining a previously generated image of an object.
You are given the following inputs:
- the StructMap image itself
- multi-view renderings of the StructMap
- the normal map of the StructMap
- the position map of the StructMap
- a textual structural description of the object
- the previously generated image
Your task is to regenerate the image based on the previous generated result while strictly enforcing structural consistency with the StructMap.
Requirements:
1. Preserve the topology, geometric composition, spatial arrangement, and relative size proportions defined by the StructMap.
2. Correct any structural inconsistencies that appear in the previous generated image.
3. Maintain smooth connections and reasonable geometric transitions between components.
4. Preserve useful visual details from the previous image whenever possible.
5. Keep the camera viewpoint consistent with the StructMap.
The regenerated image should faithfully follow the structural constraints while remaining visually realistic.

Detail Enhancement Prompt (in Detail Enhancement Skill). This prompt is used when the orchestration core determines that the generated image lacks sufficient geometric details. The goal of this prompt is to enrich geometric details while preserving the structural layout defined by the StructMap. Since the notion of geometric details can be ambiguous for MLLMs, we adopt an in-context learning strategy by providing illustrative examples of common geometric details, as illustrated in Fig. 19.

You are tasked with generating a photorealistic image of an object based on a StructMap.
The StructMap defines the object's topology, geometric composition, spatial

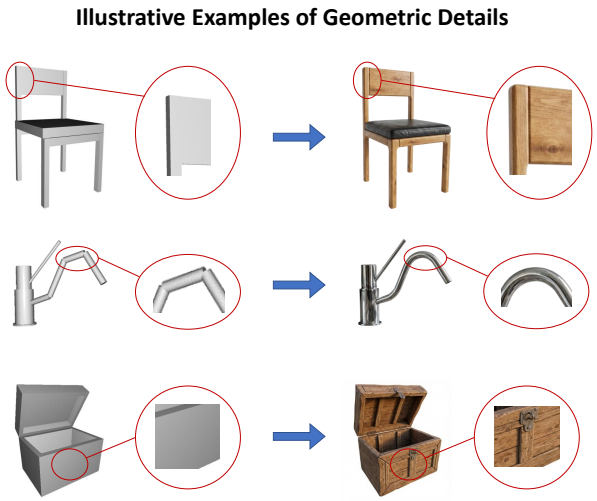

Figure 19: Illustrative examples of common geometric details for the Detail Enhancement Skill.

arrangement, and relative size proportions. The current generated image already follows the correct structural layout.
However, the StructMap lacks fine-grained geometric details such as edge smoothing, bevels, natural transitions between components, and subtle surface variations that commonly appear in real-world objects.
Your task is to enhance the geometric realism of the object by introducing appropriate geometric details while preserving the original structure.
To guide the generation process, you are provided with illustrative examples of geometric details. These examples demonstrate how basic geometric primitives can be refined into more realistic forms by adding:
- smoothed edges and bevels
- natural transitional connections between parts
- refined junctions and attachment regions
- subtle surface geometric variations
Requirements:
1. Preserve the topology, spatial layout, and relative size proportions defined by the StructMap.
2. Maintain the overall structure of the previously generated image.
3. Introduce realistic geometric refinements that improve visual plausibility.
4. Avoid altering the main structural components or camera viewpoint.

Appearance Attribute Inference Prompt (in Appearance Refinement Skill). This prompt is used to infer plausible appearance attributes when the generated image lacks realistic textures or materials.

You are a visual appearance reasoning assistant for 3D objects.
You are given an image of an object generated from a StructMap. The structural layout of the object is already correct, but the texture and material appearance may not be realistic.
Your task is to infer plausible texture and material attributes for different parts of the object.
When making your inference, consider:
- the possible category of the object
- the functional role of each component
- the geometric characteristics of each part (e.g., flat surface, cylindrical body, supporting base)
For each major component of the object, describe:
1. the likely material type
2. the texture characteristics

3. the surface finish (e.g., matte, glossy, metallic)
4. any plausible visual details such as subtle patterns, reflections, or wear.
Focus only on appearance attributes. Do not modify the structure.

Texture and Material Description [Model Response] (in Appearance Refinement Skill). Below we show an example of the texture and material attributes inferred by the agent. The description summarizes plausible appearance properties for different structural components of the object.

Material Attributes:
- back: wood
- leg: stainless steel
- seat: foam cushion with fabric covering
Texture Attributes:
- back: smooth wooden grain texture
- leg: brushed metallic texture
- seat: soft woven fabric texture
Color Attributes:
- back: matte white paint on wood
- leg: silver metallic
- seat: light gray fabric
Surface Finish:
- back: matte finish
- leg: semi-gloss metallic finish
- seat: soft matte fabric finish

Regeneration Prompt (in Appearance Refinement Skill). After the appearance attributes are inferred, the agent regenerates the image using these attributes as guidance. The regeneration process is conditioned on the previously generated image to maintain useful visual details.

You are refining the appearance of a previously generated object image.
You are given:
- the StructMap image
- the previously generated image
- a description of plausible texture and material attributes for different object parts
Your task is to regenerate the image while preserving the existing structure and geometry, but improving the visual appearance according to the provided material and texture descriptions.
Requirements:
1. Preserve the topology, geometry, and spatial layout defined by the StructMap.
2. Maintain the structural correctness of the previously generated image.
3. Apply the inferred material and texture attributes to the corresponding object parts.
4. Ensure the materials, lighting, and reflections are visually realistic.
5. Improve the overall visual coherence and aesthetic quality.

Realism Description [Model Response]. Below we show an example of a realism description produced by the agent. The description records the geometric refinements and appearance attributes introduced during synthesis.

Part 1: This part forms the main cylindrical body of the object.
It has smooth rounded edges and subtle bevels at the top and bottom boundaries.
The surface is made of brushed stainless steel with fine linear texture patterns running vertically.
The material exhibits a semi-gloss metallic finish with soft reflections.
Part 2: A rectangular box attached to the side of the main body.
The connections between this component and the cylinder are smoothly blended

with rounded transitions rather than sharp edges. The surface material appears to be matte-coated plastic with a smooth, slightly micro-rough texture.
Part 3: The top cap is dome-shaped with gently curved geometry.
Its surface is polished aluminum with a glossy metallic finish, producing clear specular highlights.
Global Attributes:
The object follows a modern industrial design style.
Materials combine brushed metal and matte-coated components for visual contrast.
Lighting produces soft reflections on metallic surfaces while maintaining balanced shading across the object.

Edit Prompt. This prompt is used when users perform viewpoint or structural editing by modifying the StructMap. The purpose of this prompt is to regenerate the object image according to the new structural configuration while preserving previously established geometric details and texture-material attributes.

You are editing a previously generated object image according to a new structural condition.
You are given:
- the new StructMap image that specifies the updated viewpoint or structural configuration
- the previously generated image
- a realism description that records geometric details and texture-material attributes from previous synthesis
Your task is to generate an updated image that follows the new StructMap while preserving the previously established visual attributes.
Requirements:
1. Follow the topology, spatial layout, and viewpoint defined by the new StructMap.
2. Preserve the geometric refinements described in the realism description, such as edge smoothing, bevels, rounded transitions, and refined junctions.
3. Maintain the same texture, material properties, surface finish, and color attributes recorded in the realism description.
4. Ensure that the visual appearance remains consistent with the previous generated image.
5. Update the geometry only where required by the new StructMap while keeping other parts unchanged.
The edited image should reflect the updated structure or viewpoint while maintaining consistent geometric details and appearance attributes.

A.10 THE COMPLEXITY OF STRUCTMAPS IN OUR DATASET

To more comprehensively illustrate the complexity of StructMaps in our dataset and the relationship between complexity and creation difficulty, we present in Fig. 20 the distribution of the number of primitives in our StructMaps, as well as the average creation time for StructMaps with different primitive counts. From Fig. 20, we can observe that our dataset contains StructMaps with a wide range of complexities, and most StructMaps consist of 7-13 primitives. And the creation time of a StructMap is approximately linearly correlated with the number of primitives it contains. Moreover, we provide several representative examples of StructMaps with different numbers of primitives in Fig. 21, offering a more intuitive illustration of how the StructMaps' complexity varies. In addition, we compute the distribution of the number of topological holes in the StructMaps and present the results in Tab. 6, illustrating the diversity of topology of the StructMaps in our dataset.

Table 6: The distribution of the number of topological holes in the StructMaps.

Number of Topological Holes	0	1	2	3	4	≥ 5
Percentage (%)	28.4	23.1	16.8	18.9	8.5	4.3

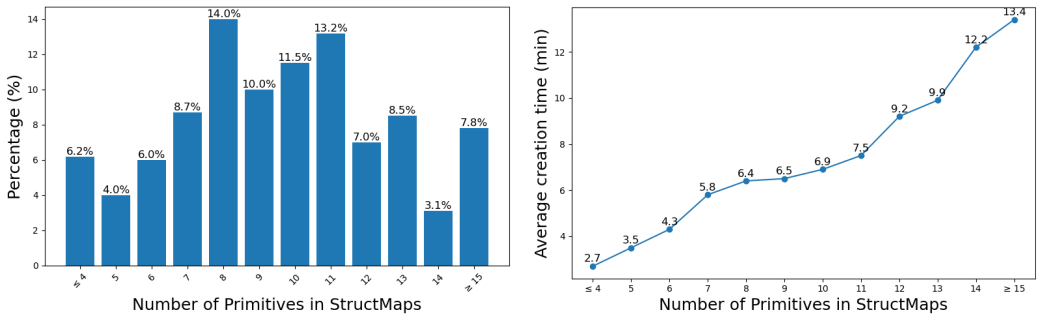

Figure 20: **[Left]** Distribution of the number of primitives in StructMap in our dataset. **[Right]** Average creation time for StructMaps with different numbers of primitives.

number of primitives	<5	5-6	7-8	9-10	11-12	13-14	>15
StructMap							

Figure 21: Representative examples of StructMaps with different numbers of primitives.

A.11 Number of Iterations in SAGA, API Response Time and Failure Cases

To further illustrate the practical behavior of our algorithm, in Fig. 22-Left we present the distribution of the iteration counts of SAGA when generating images using our dataset (We set the maximum number of iterations to 5). As shown in Fig. 22-Left, the vast majority of examples converge in 1-3 iterations, and the overall convergence success rate reaches 99.7%. Furthermore, Tab. 7 reports the average number of times each skill is invoked and the average API response time for each skill in SAGA.

Table 7: The average API response time for each module of our image generation algorithm.

Agent Skills	Number of Skill Invocations	API Response Time (s)
Structure Preservation Skill	0.76	72.24
Detail Enhancement Skill	1.33	32.76
Appearance Refinement Skill	0.69	48.35

In addition, Fig. 22-Right presents several failure cases. For example, in (a), the small sphere at the lower end of the globe's axis is omitted during generation, and in (b), the cabinet's rotation angle is incorrectly produced. These issues may arise because certain components in the image are too small or the overall layout is overly cluttered. We plan to further improve the performance of our method in future work.

A.12 Generating Other Categories of Images

To further showcase the broader potential of our method, we conducted additional generation experiments, including: 1) generating non-rigid objects, and 2) generating multi-object scenes. Specifically, for non-rigid objects, we deform a rigid StructMap according to physically plausible rules in Blender to obtain StructMaps that reflect flexible shapes; and for multi-object scenes, we create the structural condition by placing the StructMaps of different objects at their intended positions in the scene. Once these StructMap conditions are created, we feed each of them into SAGA to synthesize the corresponding photorealistic images. We present several examples of these results in Fig. 23.

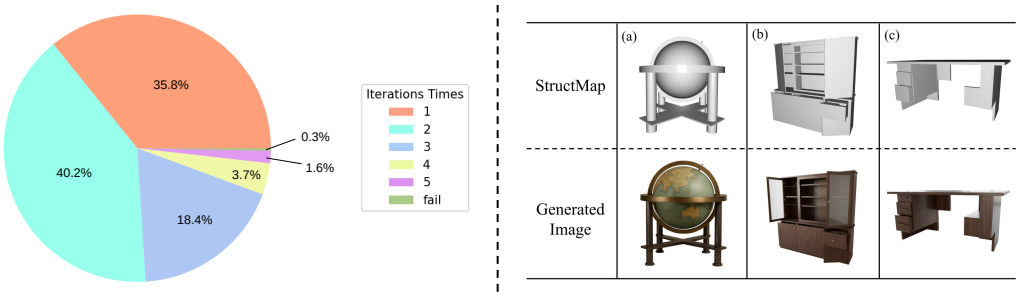

Figure 22: **[Left]** Distribution of the iteration counts of SAGA. **[Right]** Several failure cases.

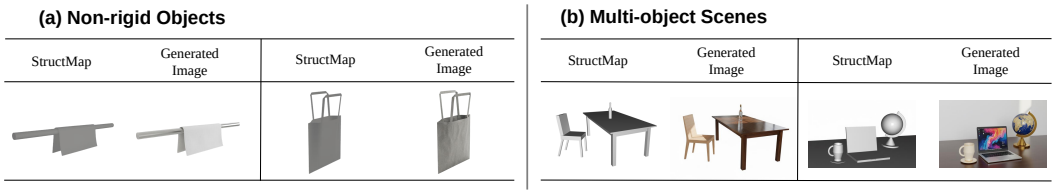

Figure 23: Examples of generating other object types, including: 1) generating non-rigid objects, and 2) generating multi-object scenes.

A.13 LIMITATION

Despite the effectiveness of Struct2Real, our work still has several limitations. First, the performance of SAGA is inevitably influenced by the capability of the underlying MLLM, especially in cases involving very small components, cluttered layouts, or subtle viewpoint changes. As shown in our failure cases, some small parts may be omitted and certain rotation angles may be inaccurately generated. Then, while we evaluate structural fidelity using both automatic metrics and human studies, existing automatic metrics are still not specifically designed for topology and spatial layout alignment. Developing more reliable and fine-grained structural evaluation metrics remains an important direction for future work. Finally, our experiments mainly focus on object-centric image generation, and extending Struct2Real to more diverse scenarios, such as complex multi-object scenes and broader non-rigid categories, is left for future exploration.