FreeInsert: Disentangled Text-Guided Object Insertion in 3D Gaussian Scene without Spatial Priors

Chenxi Li
Tianjin University
Tianjin, China
chenxili2024@tju.edu.cn

Weijie Wang†
University of Trento
Trento, Italy
weijie.wang@unitn.it

Qiang Li
Tianjin University
Tianjin, China
liqiang@tju.edu.cn

Nicu Sebe
University of Trento
Trento, Italy
niculae.sebe@unitn.it

Bruno Lepri
Fondazione Bruno Kessler
Trento, Italy
lepri@fbk.eu

Weizhi Nie
Tianjin University
Tianjin, China
weizhinie@tju.edu.cn

Abstract

Text-driven object insertion in the 3D scene is an emerging task that enables intuitive scene editing through natural language. Despite its potential, existing 2D editing-based methods often suffer from reliance on spatial priors such as 2D masks, 3D bounding boxes, and they struggle to ensure inserted object consistency. These limitations hinder flexibility and scalability in real-world applications. In this paper, we propose *FreeInsert*, a novel framework that leverages foundation models (MLLMs, LGM, and diffusion models) to disentangle object generation and spatial placement, enabling unsupervised and flexible object insertion in 3D scenes without spatial priors. *FreeInsert* begins with an MLLM-based parser that extracts structured semantics—including object types, spatial relationships, and attachment regions—from user instructions. These semantics guide both the reconstruction of the inserted object for 3D consistency and the learning of its degrees of freedom. We first leverage the spatial reasoning capabilities of MLLMs to initialize the object's pose and scale. To further enhance natural integration with the scene, a hierarchical spatially-aware stage is employed to refine the object's placement, incorporating both the spatial semantics and priors inferred by the MLLM. Finally, the object's appearance is enhanced using inserted-object image to improve visual fidelity. Experimental results demonstrate that *FreeInsert* enables semantically coherent, spatially precise, and visually realistic 3D insertions, without requiring any spatial priors, offering a user-friendly and flexible editing experience. Project page: https://tjulcx.github.io/FreeInsert/.

CCS Concepts

• **Computing methodologies** → **Computer vision**.

Weijie Wang† is the Corresponding author.

MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755072

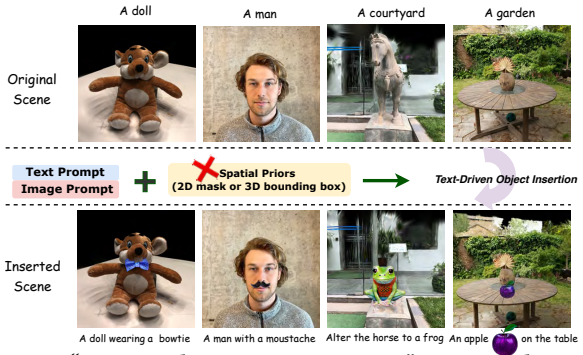

Figure 1: "No Spatial Priors, Just Prompts." Compared to existing methods that require user-provided spatial priors, limiting their practicality, our method enables flexible text-driven object insertion without any need for such priors (e.g., 2D masks or 3D bounding boxes). Given only a text prompt (The image prompt is optional), *FreeInsert* naturally inserts objects across diverse scenes.

Keywords

Text-Driven 3D Scene Editing; Object Insertion; Diffusion Models; Multimodal Large Language Models; Gaussian Splatting

ACM Reference Format:
Chenxi Li, Weijie Wang†, Qiang Li, Nicu Sebe, Bruno Lepri, and Weizhi Nie. 2025. FreeInsert: Disentangled Text-Guided Object Insertion in 3D Gaussian Scene without Spatial Priors. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3755072

1 Introduction

Text-driven 3D generation [3, 24, 28, 40, 47] and editing [2, 43, 46] are gaining traction for enabling the intuitive customization of digital content with a few words. Despite recent advances [9, 10, 12, 17, 23, 25, 32, 37–39, 49] that have made significant progress in editing the geometry and appearance of scene components, flexibly inserting new objects into the scene remains challenging due to difficulties in precise placement and seamless integration.

Recent 3D editing methods leverage diffusion models by first performing text-guided 2D edits on single [6, 30] or multi-view images [12, 49], then lifting them to 3D. Relying solely on textual descriptions for object insertion often leads to insertion failure and suboptimal results due to misinterpretation of the text [48,

49], as shown in Figure 3, exemplified by methods like Instruct-NeRF2NeRF [12] and GaussCtrl [41]. Some methods introduce attention mechanisms to capture the spatial relationship between the inserted object and the scene. However, these methods still struggle with accurately determining the inserted object's pose and scale [33, 46, 49]. To address this limitation, other methods leverage user-provided 2D masks [6, 30] or 3D bounding boxes [31, 48] as strong constraints to achieve more controllable and concise insertion. Nevertheless, they often demand specialized expertise [48] and considerable manual effort, limiting their practical usability. In addition, they still face challenges with inaccurate depth estimation [6] and inconsistent 3D multi-view reconstruction due to the modality gap between 2D and 3D. We summarize the above discussion as shown in Table 1.

Inspired by this, achieving flexible and high-quality object insertion into 3D scenes without manual supervision remains underexplored. The advent of large-scale models [1, 8, 11, 21, 42], which have acquired human commonsense knowledge, has made unsupervised learning increasingly promising. In this paper, we propose *FreeInsert*, a method that leverages foundation models (MLLMs [1, 8], LGM [34] and Diffusion model [29]) to assist object insertion in 3D scenes without relying on any spatial priors as Figure 1. Our method removes the need for spatial priors by inferring object insertion directly from high-level textual cues (e.g., *"Add [object] to/on [target]"*) as Figure 1 shows. We argue that **the insertion process can essentially be viewed as first generating a object, followed by estimating the transformation, which defines the inserted object's degrees of freedom (pose and scale) relative to the scene.**

Specifically, we disentangle the object insertion process into object generation and its parameterized degrees of freedom (DoF) estimation, both guided by textual descriptions with foundation models. We first obtain a text instruction from the user and parse it into structured semantics (e.g., object type, spatial relation, attachment region) using an MLLM-based [1] object insertion parser. This enables precise, controllable object insertion that aligns with the user's intent. We then employ a 3D-consistent reconstruction model [34] to obtain an initial Gaussian-based object model, which is coarsely inserted into the scene guided by the visual and spatial reasoning capabilities of MLLMs[1, 8]. This step inherently circumvents the inconsistency issues associated with 2D editing-based methods. While the feed-forward procedure provides a lightweight 3D layout, it often suffers from suboptimal placement and imperfect geometry. To address these issues, we propose a two-stage refinement. First, the Hierarchical Spatial-Aware Refinement stage optimizes the object's DoF via spatially-aware score distillation sampling (SSDS) [7] from pretrained diffusion model [29]. This stage leverages MLLM-derived reasoning results to align the object's DoF with both local and global spatial semantics, enhancing more precise and controllable placement. These reasoning results also help the model handle rare spatial composition e.g., "*Add a pair of sunglasses on the forehead*", thereby improving robustness. In the final appearance refinement stage, we fine-tune a pretrained diffusion model on multi-view renderings of the optimized inserted object and its corresponding inserted-object image, and use it to enhance the object's appearance. By disentangling placement from object generation, our method enables flexible semantic control over insertion while preserving object quality and ensuring coherent, plausible 3D scenes.

Table 1: Comparison of existing methods of object insertion in 3D scenes. Ours can achieve high-quality object insertion without manual supervision while keeping 3D consistency.

	No Required Manual Supervision	3D View Consistency	Support Image-Prompts
Instruct-N2N[12]	✓	✗	✗
GaussCtrl [41]	✓	✓	✗
GaussianEditor [6]	✗	✗	✗
TIP-Editor [48]	✗	✗	✓
FreeInsert	✓	✓	✓

To evaluate the proposed method, we applied it to various scenarios, including object-centric, human-centric, and complex outdoor scenes. Our experimental results demonstrate that the proposed approach can insert diverse objects into 3D scenes without requiring manual supervision while achieving multi-view consistent object quality. In summary, our contributions are as follows:

- We address consistent object insertion in diverse 3D scenes using only textual input, removing the need for spatial priors and outperforming existing methods through a framework that disentangle object generation and spatial placement.
- We propose a DoF optimization method for object insertion, using the reasoning capabilities of MLLMs and diffusion models in place of manual supervision. The MLLM's semantic and spatial priors further support SSDS in enhancing precision and robustness.
- We ensure high-quality object generation by maintaining 3D shape consistency via a reconstruction model and refining visual appearance.
- We present the first baseline for evaluating unsupervised 3D scene insertion, with experiments showing competitive performance against state-of-the-art methods.

2 Related Works

Text-Guided 3D Scene Editing. Text-driven 3D scene editing has seen rapid progress, thanks to the rise of diffusion models [13, 29]. Most methods [9, 12, 18, 19, 23, 27, 36] focus on modifying existing content, either globally or locally. Local editing requires precise localization to avoid affecting unrelated regions, which remains challenging. While some works use implicit cues from models like InstructPix2Pix [4] or ControlNet [44], others incorporate explicit constraints such as segmentation masks [37] or cross-attention maps [49]. However, these methods struggle with 3D object insertion, which demands reasoning about semantically appropriate yet physically unoccupied regions for placement. In this work, we primarily focus on the task of object insertion in 3D scenes.

Object Insertion in 3D Scene. In contrast to modifying existing scene content, object insertion remains underexplored. MVInpainter [5] leverages segmentation to identify support regions like table surfaces, but struggles with fine-grained insertions on objects or humans. GaussianEditor [6] and InseRF [30] insert objects using 3D reconstruct models, yet still require user-provided 2D masks and suffer from depth-related localization issues. FocalDreamer [20] attaches parts to base shapes but depends on user-specified 3D parameters (e.g., rotation, translation, scale), and lacks generalization to complex scenes. Other methods [31, 48] guide object generation

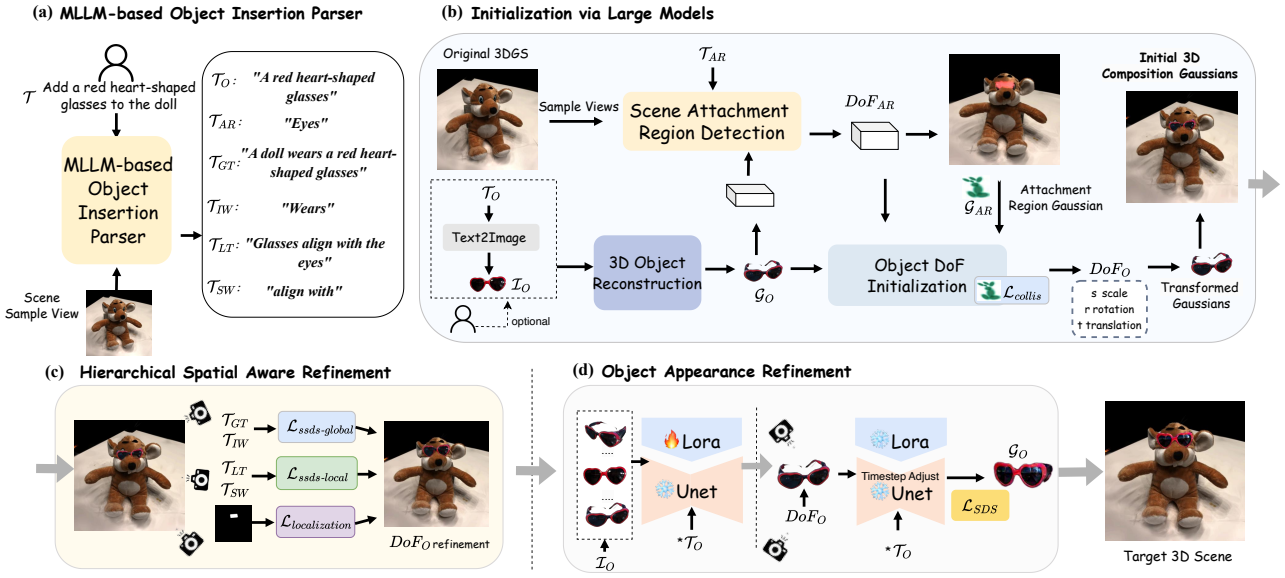

Figure 2: Overview of *FreeInsert*. Given an text prompt $\mathcal{T}$ and optionally an image prompt $\mathcal{I}_O$, the object insertion process includes four stages: (a) The MLLM-based Object Insertion Parser (see Section 3.2) first extracts structured semantics to support the subsequent stages. (b) The Initialization via Large Models (see Section 3.3) stage generates object and initializes its DoF_O in the scene . (c) The Hierarchical Spatial Aware Refinement (see Section 3.4) stage refines the DoF_O. (d) The final stage, Object Appearance Refinement (see Section 3.5), enhances the object's visual quality using object image$\mathcal{I}_O$.

via diffusion using 3D bounding boxes, which imposes a burden on users. In this work, we aim for unsupervised and broadly applicable 3D object insertion, removing the need for manual annotations or spatial priors.

Large Language Models in 3D Generation and Editing. LLMs, like GPT [1] and Llama [11] series, have exhibited outstanding efficacy in many text-related tasks. Zhou et al. [47] and Zhou et al. [45] utilize LLMs to provide coarse compositional spatial information from textual descriptions to construct the 3D scene. The multi-modal variants of LLMs [1, 8] incorporate images and are additionally trained on image-text pairs, showing impressive results for visual captioning and vision question-answering (VQA). Notably, Molmo [8] can perform pixel-level localization mainly because it was trained with richly annotated image data. This capability is crucial for robust spatial grounding in vision-language tasks. GG-Editor [43] first exploits GPT-4V [1] to better understand both the textual and 3D visual inputs and then infer reasonable local regions for 3D editing. However, it primarily targets object editing and its preliminary use of MLLM struggles to ensure spatial precision. In this work, we leverage the text reasoning capabilities and spatial relationship understanding of GPT-4 [1] and Molmo [8], supplemented by a basic detection model [42], to eliminate the reliance on manually provided priors in object insertion.

3 Method

3.1 Problem Statement

Figure 2 illustrates the overall framework of *FreeInsert*. Given a group of 3D Gaussians $\mathcal{G}_S$ for an input scene and a text prompt $\mathcal{T}$ guiding the insertion of an object into the scene, our algorithm performs high-quality, semantically consistent object insertion *without any manual supervision (e.g., 3D bounding boxes or masks).* We decouple the object insertion task into object generation and the optimization of the object's 3D degrees of freedom DoF_O (rotation, translation, scale), both guided by semantic alignment between the resulting scene and the user prompts. The resulting scene is represented by a new set of Gaussians, $\mathcal{G}_{inserted}$. Moreover, our method allows image prompt $\mathcal{I}_O$ as input to specify the object's appearance. Formally, the insertion process is defined as:

$$\mathcal{G}_{inserted} = \mathcal{E}\left(\mathcal{G}_S,\ \mathcal{T},\ \mathcal{I}_O\right), \tag{1}$$

where $\mathcal{E}$ denotes the process applied to the inserted object, including its generation, DoF learning in the context of the scene $\mathcal{G}_S$, and appearance refinement.

3.2 MLLM-based Object Insertion Parser

A key challenge in unsupervised object insertion is converting high-level user intent into structured, fine-grained guidance. To address this, we introduce an MLLM-based Object Insertion Parser (MLLM-OIP) that utilizes the MLLM's spatial understanding capability to parse the instruction $\mathcal{T}$ into semantically prompts, providing essential guidance for the subsequent object insertion. Specifically, we provide a prompt template $\mathcal{T}_{parser}$ and a sampled scene image $\mathcal{I}_S$ as input to the multimodal LLM $\mathcal{M}_{MLLM}$ [1] to obtain structured outputs. The prompts generation process is formalized as :

$$(\mathcal{T}_O,\ \mathcal{T}_{AR},\ \mathcal{T}_{GT},\ \mathcal{T}_{IW},\ \mathcal{T}_{LT},\ \mathcal{T}_{SW}) = \mathcal{M}_{MLLM}\left(\mathcal{T},\ \mathcal{T}_{parser},\ \mathcal{I}_S\right) \tag{2}$$

Here, the Object Prompt ($\mathcal{T}_O$) is used for 3D object generation and appearance refinement stage. The Attachment Region Prompt($\mathcal{T}_{AR}$)

plays a crucial role in the initialization of the object's degrees of freedom DoF_O. The remaining four prompts including the Global Target Prompt ($\mathcal{T}_{GT}$) and its Object Interaction Word ($\mathcal{T}_{IW}$), the Local Target Prompt ($\mathcal{T}_{LT}$) and its Spatial Relationship Word ($\mathcal{T}_{SW}$) are employed during the hierarchical spatial-aware refinement stage to refine the DoF_O, supporting global-local semantic alignment.

3.3 Initialization via Large Models

Object from Prompts. To avoid 3D inconsistency, we first use a text-to-image (T2I) [29] model to synthesize a Text-generated image $\mathcal{I}_O$ of the object from the object description prompt $\mathcal{T}_O$. The synthesized image is then used to recover the 3D geometry $\mathcal{G}_O$ via LGM [34], a single-view reconstruction model that achieve a trade-off between reconstruction quality and efficiency. Other lightweight 3D reconstruction methods [14, 22] can also be adopted. In addition, $\mathcal{I}_O$ can be directly specified by the user, allowing for more precise control over the object's appearance.

Scene's Attachment Region Detection. Intuitively, an object's placement is influenced by the attachment region of the scene and the degrees of freedom within that region. Driven by this, we extract an attachment region $\mathcal{G}_{AR}$ and its associated degrees of freedom $DoF_{AR}(s_{AR}, r_{AR}, t_{AR})$ from the 3D scene based on the Attachment Region Prompt $\mathcal{T}_{AR}$. This region serves as a crucial spatial reference, guiding initializing the inserted object DoF_O. Specifically, we employ an open-vocabulary detection model Florence2 [42] to localize 2D bounding boxes across smapled views from the scene with camera poses C_{cam}, guided by $\mathcal{T}_{AR}$. For each view, the detected box is converted into a binary mask $\mathcal{I}_{BAR}$, representing the candidate attachment region. The 3D attachment area is parameterized by the degrees of freedom of a initial 3D bounding box $\mathcal{B}_{init}$. We optimize the attachment by computing cross-entropy between the projected transformed bounding box and the detected attachment region mask $\mathcal{B}_{\text{AR}}$ across all camera views. Thus, DoF_{AR} is calculated as follows:

$$DoF_{AR} = \arg\min_{\theta} \sum_{\mathcal{T}_{cam} \in C_{cam}} \mathcal{L}_{BCE}\left(Proj\left(\mathcal{B}, \mathcal{T}_{cam}\right), \mathcal{I}_{BAR}^{(\mathcal{T}_{cam})}\right), \quad (3)$$

where $\theta = (s, r, t)$ denotes the transformation parameters for the canonical box $\mathcal{B}_{init}$, $\mathcal{F}_{affine}$ is the affine transformation function, and the transformed 3D bounding box is computed as $\mathcal{B} = \mathcal{F}_{affine}(\mathcal{B}_{init}, \theta)$. The function $\text{Proj}(\mathcal{B}, \mathcal{T}_{cam})$ denotes the 2D projection of the 3D bounding box $\mathcal{B}$ onto the image plane under the camera pose $\mathcal{T}_{cam}$. After obtaining DoF_{AR}, the attachment region $\mathcal{G}_{AR}$ is extracted by selecting 3D Gaussians from the scene representation $\mathcal{G}_S$ within the transformed bounding box $\mathcal{B}_{AR} = \mathcal{F}_{affine}(\mathcal{B}_{init}, DoF_{AR})$. Formally, the attachment region is defined as:

$$\mathcal{G}_{AR} = \{g \in \mathcal{G}_s \mid g \in \mathcal{B}_{AR}\}$$

Object's DOF Initialization. Once obtained the attachment region $\mathcal{G}_{AR}$ and its associated transformation DoF_{AR}, we initialize the inserted object's degrees of freedom DoF_O (s_O, r_O, t_O) accordingly. For the s_O initialization, we assume an intuitive real-world prior: there exists a reasonable relative scale ratio λ_{rel} between the inserted object and the attachment region, which helps ensure a plausible insertion. This ratio is implicitly understood by large-scale language models. Therefore, we leverage $\mathcal{M}_{MLLM}$ to predict λ_{rel}, and compute the object scale as $s_O = s_{AR} \cdot \lambda_{rel}$. Considering the uncertainty in MLLM predictions and the influence of scale initialization quality on subsequent refinement, we adopt an iterative strategy. After initializing r_O and t_O, we render the scene with the inserted object and iteratively interact with the MLLM, using visual feedback to adjust s_O and improve realism and integration.

For the r_O initialization, we leverage $\mathcal{M}_{MLLM}$ to initialize a semantically appropriate object rotation. Given a or MLLM-suggested primary scene viewpoint, we render a scene image $\mathcal{I}_S$, and sample a set of object-centric renderings $\{\mathcal{I}_O^{(r)}\}_{r \in \mathcal{R}}$ for the inserted object, where each $r \in \mathcal{R}$ corresponds to a unique azimuth-elevation rotation $(\phi, \theta) \in [0, 2\pi) \times [0, \pi)$. Based on the $\mathcal{I}_S$, the rendering set $\{\mathcal{I}_O^{(r)}\}$, and the Global Target Prompt $\mathcal{T}_{GT}$, the model can select the optimal rotation r_O that maximizes a semantic alignment score:

$$r_O = \arg\max_{r \in \mathcal{R}} \mathcal{M}_{MLLM}(\mathcal{I}_S, \mathcal{I}_O^{(r)}, \mathcal{T}_{GT}) \quad (4)$$

where $\mathcal{M}_{MLLM}$ evaluates sementic plausibility the placement aligns with the scene.

To initialize the t_O, we use strong pixel-level semantic spatial localization capability of Molmo [8] to predict a set of 2D object centers $\{c_O^{(v)}\}_{v \in \mathcal{V}}$ across multiple scene views with the Local Target $\mathcal{T}_{LT}$, using prompt like "*Point the position to add <$\mathcal{T}_{LT}$>*". Let $\hat{\mathcal{G}}_O^{(t)}$ denote the object geometry after applying the transformation (s_O, r_O, t), where t is a optimized parameter. For each view v, we project the transformed object and compute the 2D centroid of its projection. The t_O is obtained by optimizing t to minimize the discrepancy between the projected and the predicted centroids:

$$t_O = \arg\min_{t} \sum_{v \in \mathcal{V}} \left\| Centroid\left(\pi_v\left(\hat{\mathcal{G}}_O^{(t)}\right)\right) - c_O^{(v)} \right\|_2^2 + \mathcal{L}_{\text{coll}}(\mathcal{G}_{\text{AR}}, O_c) \quad (5)$$

Here, $\pi_v(\cdot)$ is the camera projection function for view v, and *Centroid (·)* computes the 2D projected center of the object. To ensure physical plausibility during object insertion, we introduce the collision loss [45] $\mathcal{L}_{\text{coll}}$, which penalizes interpenetration between the object centriod O_c and the scene attachment region $\mathcal{G}_{\text{AR}}$.

3.4 Hierarchical Spatial Aware Refinement

The initial DoF_O from $\mathcal{M}_{MLLM}$ often lack spatial accuracy, hindering seamless scene integration. Base on that, we then optimize the DoF_O using SSDS Loss [7], refining the object's placement in the scene. The loss is defined as:

$$\nabla_\theta \mathcal{L}_{\text{SSDS}}(\phi^\star, x) = \mathbb{E}_{t,\epsilon}\left[w(t)\left(\hat{\epsilon}_{\phi^\star}(x_i; y, t) - \epsilon\right) \frac{\partial x}{\partial \theta} \right] \quad (6)$$

Here, θ, x, ϕ^*, and $\hat{\epsilon}_{\phi^*}(x_t; \mathcal{T}, t)$ denote the 3D representation, rendered image, spatial attention map, and the score function predicting noise ϵ from the noised image x_t with text prompt $\mathcal{T}$. Unlike the original design for multi-object composition with high timesteps, we find that lower timesteps are more effective for fine-grained DoF refinement in our setting, as it emphasizes local spatial details critical for precise alignment.

Global-Local Collaborative Spatial Awareness. Diffusion models often exhibit spatial biases due to training data imbalance, e.g., generating moustaches at a relatively large scale across the lower face. Large models trained on data-driven priors often fail to meet human expectations in spatial reasoning. (see Figure 6) To address spatial ambiguity, we leverage spatial relation terms (e.g., "on", "in front of") to impose explicit constraints on object localization.

Compared to general verbs like "wearing" or "with", these relations encode more precise spatial priors, leading to more effective supervision for optimizing object placement. We leverage spatial prompts inferred from MLLM-OIP, which offers both global semantic grounding $\mathcal{T}_{GT}$ with interaction word$\mathcal{T}_{IW}$ and fine-grained positional cues $\mathcal{T}_{LT}$ with spatial relationship word$\mathcal{T}_{SW}$. We define a hierarchical spatial loss that jointly supervises local and global alignment:

$$\mathcal{L}_{\text{spatial}} = \beta \cdot \mathcal{L}_{ssds\text{-}global}(\mathcal{T}_{GT}, \mathcal{T}_{IW}) + (1-\beta) \cdot \mathcal{L}_{ssds\text{-}local}(\mathcal{T}_{LT}, \mathcal{T}_{SW}), \quad (7)$$

Attention-based Localization. We found that due to the bias in the T2I model's training data, SSDS Loss exhibits limitations when handling rare spatial relationships. These inherent limitations restrict the effectiveness of prompt instructions. To enable stronger spatial conditioning, we adopt attention-based localization loss [48], enforcing tighter regional constraints as follows:

$$\mathcal{L}_{loc} = \left(1 - \max_{s \in \mathcal{S}}(A_t^s)\right) + \lambda \sum_{s \in \tilde{\mathcal{S}}} \left\|A_t^s\right\|_2^2 \quad (8)$$

where λ balances the two terms, $\mathcal{S}$ denotes the multi-view mask region projected from the 3D bounding box $\mathcal{B}$, obtained by tightly enclosing the object after DoF initialization, and $\tilde{\mathcal{S}}$ denotes the complementary region. As shown in our ablation (Figure 7), it is essential for precise object placement within the designated area.

3.5 Object Appearance Refinement

Once the object's degrees of freedom are determined, a refinement module is introduced to enhance the visual quality of the inserted object $\mathcal{G}_O$. Specifically, we refine $\mathcal{G}_O$ using the high-quality appearance from the inserted-object image $\mathcal{I}_O$ via LoRA [15].

Viewpoint Frequency Balancing. To avoid the overfitting caused by using a single-view optimization, which often leads to 3D inconsistencies and missing object parts (e.g., a side view causing missing legs) as shown in Figure 9 (b). We perform multi-view sampling of the inserted object. Specifically, given a set of views $\{I_i\}_{i=1}^N$, rendered from the inserted object $\mathcal{G}_O$. We estimate the pose P^* of the object image $\mathcal{I}_O$ by selecting the most similar view based on DINO feature similarity [26]. To ensure both appearance fidelity and geometric consistency, We construct the training set $\mathcal{D}_{\text{ref}}$ by combining the rendered multi-view images with repeated samples of the inserted-object image $\mathcal{I}_O$ and its estimated pose P^*, as follows:

$$\mathcal{D}_{\text{ref}} = \{(I_i, P_i)\}_{i=1}^N \cup \{(\mathcal{I}_O, P^*)\}_{j=1}^M \quad (\text{repeated},\ M > N) \quad (9)$$

The following objective is used to fine-tune the LoRA layers:

$$\mathcal{L}_{\text{ref}} = \mathbb{E}_{z_i, I_i, P_i, y^*, \epsilon, t} \left\|\epsilon_{\phi_2}(z_i, t, P_i, I_i, y^*) - \epsilon\right\|_2^2, \quad (I_i, P_i) \sim \mathcal{D}_{\text{ref}}. \quad (10)$$

z_i is the noisy latent of image I_i, t is the diffusion timestep, and ϵ is the target noise. The denoising network ϵ_{ϕ_2}, augmented with LoRA [15], is conditioned on I_i, its pose P_i, and the object-specific prompt y^*, e.g., "A <token> dog", which is formatting from $\mathcal{T}_O$.

Appearance-Focused Refinement. We employ fine-tuning diffusion to update the object Gaussian $\mathcal{G}_O$, guided by $\mathcal{L}_{sds}$ [28]. Under our setting, we sample from a lower range of timesteps during optimization to reduce the impact on the object's geometry (e.g., shape and scale), thereby encouraging the model to focus more on refining appearance details. The corresponding objective is defined as follows:

$$\nabla_\theta \hat{\mathcal{L}}_{sds}(\phi, x) = \mathbb{E}_{\hat{t}, \epsilon}\left[w(\hat{t})\left(\hat{\epsilon}_\phi(x_i; y_i, \hat{t}) - \epsilon\right)\frac{\partial x}{\partial \theta}\right], \quad (11)$$

where $\hat{t}$ denotes the adjusted (lower) diffusion timestep.

3.6 Object Replacement

In addition, our method can be naturally extended to **object replacement** in the scene. Specifically, given a user prompt such as *"Add a [new object] to replace [existing object]"*, the corresponding object to be replaced is identified through the Attachment Region $\mathcal{G}_{AR}$. We remove $\mathcal{G}_{AR}$ and then execute the standard insertion pipeline, enabling replacement without being constrained by the original object's geometry or structure.

4 Experiments

4.1 Experiments Setup.

Implementation Details During initialization, we use a learning rate of 5×10^{-3} for optimizing both $\mathcal{G}_{AR}$ and t_O of inserted object. When estimating the coarse rotation r_O, we render the object at 10-degree intervals . During the Hierarchical Spatial Aware Refinement stage, we apply a learning rate of 5×10^{-4} with diffusion timesteps in the range of [0.02, 0.2], $\lambda = 0.1$ is set in $\mathcal{L}_{loc}$ and β is linearly increased from 0 to 1 during training. For appearance refinement, we optimize the object appearance using timesteps in [0.02, 0.5~0.25]. The object image $\mathcal{I}_O$ is upsampled with a sampling ratio of $M/N = 3$ relative to multi-view inputs. All experiments are conducted on a single NVIDIA A40 GPU. More details are provided in the Appendix.

Dataset To comprehensively evaluate our method, we follow prior works [6, 12, 48] and select representative scenes of varying complexity, including simple backgrounds, human faces, and complex outdoor environments. In these scenes, we insert commonly associated objects (e.g., glasses, giraffes) and evaluate diverse categories such as bowties and moustaches to assess generalization. For GaussianEditor [6], we manually annotate masks, while for TIP-Editor [48], we use the author-provided bounding boxes and object images for comparison.

Baselines We compare our method with state-of-the-art 3D scene editing approaches that support object insertion and replacement, under two types of guidance: text prompt and text-image prompt.The text-guided baselines include three methods: Instruct-GS2GS [35], which extends Instruct-NeRF2NeRF (IN2N) [12] by replacing the NeRF in IN2N with a 3DGS model; GaussCtrl [41], and GaussianEditor [6]. For text-image prompt methods, we compare with TIP-Editor [48], which uses an example image to specify object appearance. As TIP-Editor provides only limited insertion scripts (e.g., "A doll wearing sunglasses", "A man with beard"). For fairness, we use official code and pre-trained weights.

Evaluation Criteria. We use CLIP Text-Image directional similarity following [6, 41, 48, 49] to assess the alignment between the text and the editing results. For appearance-specified cases, we further employ DINO similarity [26] following [48] to assess appearance preservation. We also conducted a user study with 50 participants,

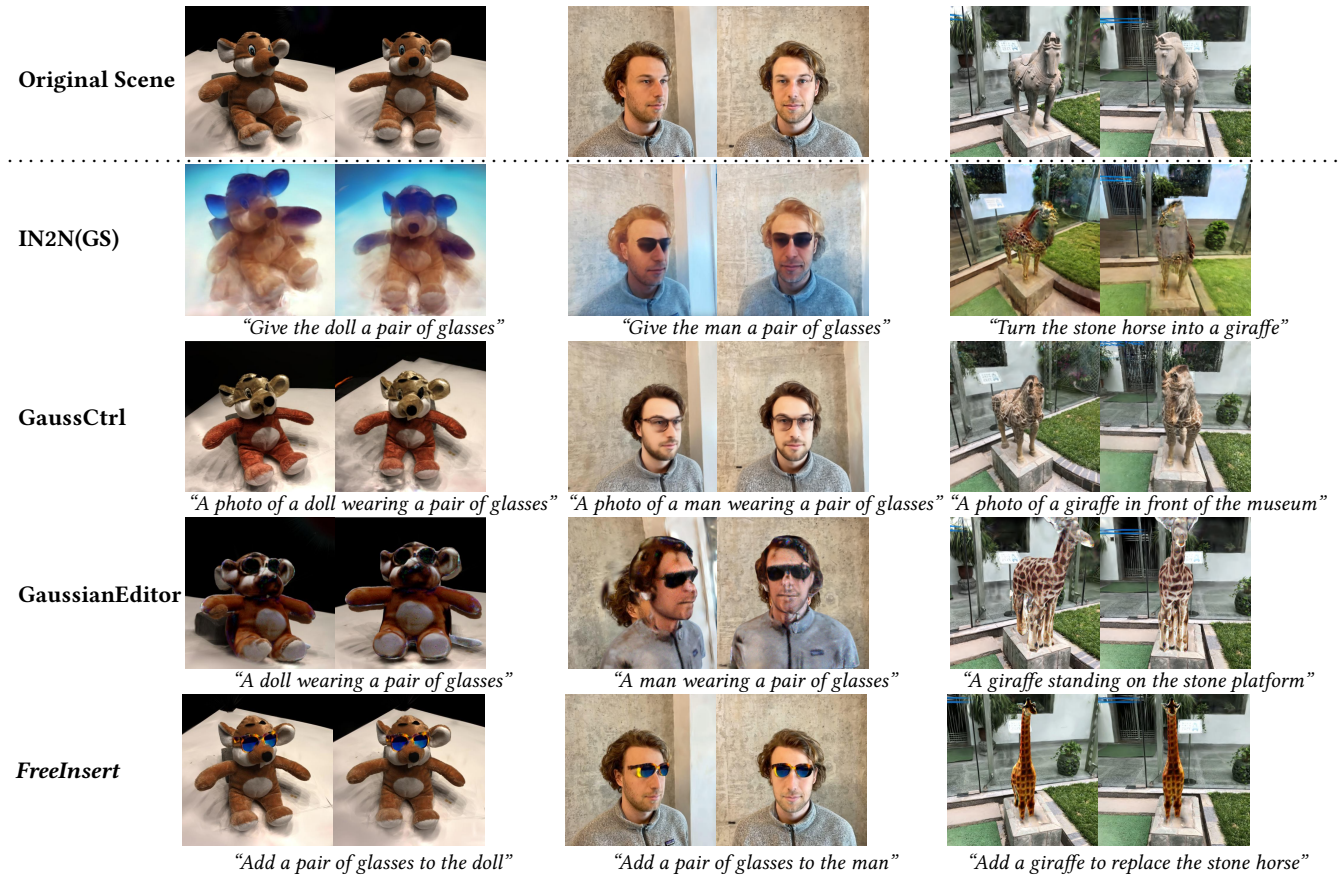

Figure 3: Visual comparison with state-of-the-art methods for text-guided object insertion (Cols 1–2) and replacement (Col 3). Our method generates higher-quality results while preserving scene integrity. IN2N (GS) and GaussCtrl sometimes misunderstand the prompt and fail to complete insertion (e.g., "Give the doll a pair of glasses to the doll"), and struggle to produce clear shape changes in replacement (Col 3, Rows 2–3). GaussianEditor requires manual masks and depth adjustment, and suffers from artifacts and low-quality objects due to post-inpainting and 3D reconstruction limitations.

who rated the 3D editing results (presented with prompts in shuffled order) on four criteria: Semantic Alignment, Object Integrity, Geometric Consistency, and Detail Preservation, using a 1–10 scale.

4.2 Comparisons with State-of-the-Art Methods

4.2.1 Qualitative comparisons. In this part, we conduct a qualitative comparison with different baselines under two types of input settings (text prompt and text-image prompt) to evaluate their performance under identical conditions. Video demonstrations are included in the supplementary.

Text Prompt Comparisons. Figure 3 shows visual comparisons between our method with three baselines. Both IN2N(GS) and GaussCtrl, which rely solely on semantic guidance, struggle to successfully complete insertions in some scene-object combinations, such as "Add a pair of glasses to the doll". Although GaussCtrl improves consistency, the replacements remain too similar to the original (e.g., a horse), failing to convincingly resemble the target (e.g., a giraffe). GaussianEditor relies on user-provided 2D masks for object insertion but struggles in object- or human-centric scenes due to inaccurate post-inpainting segmentation, leading to artifacts like foreground overlaps. While it performs better in outdoor scenes (e.g., the giraffe example), its depth estimation is often imprecise and requires manual adjustment. By contrast, our method achieves high-quality object insertion and replacement results in both scene preservation and object completeness without requiring any manual annotations.

Text-Image Prompt Comparisons. Besides the text-prompt methods, we further evaluate object insertion and replacement with a given image prompt of the specified object in Figure 4, comparing against TIP-Editor [48]. Although TIP-Editor supports flexible insertion via 3D bounding boxes, it suffers from inconsistent multi-view appearances due to its reliance on 2D editing techniques. Most critically, achieving such results remains dependent on finely user-provided 3D bounding boxes, which significantly hinders scalability and practicality. In contrast, our method delivers the most complete geometry and better appearance fidelity to the image prompt without relying on any annotation.

4.2.2 Quantitative Comparisons. Table 2 presents the quantitative comparison of our method against other baseline methods. Our method achieves CLIP text–image semantic alignment scores comparable to the state-of-the-art methods without requiring any manual annotations for the insertion region. Moreover, compared

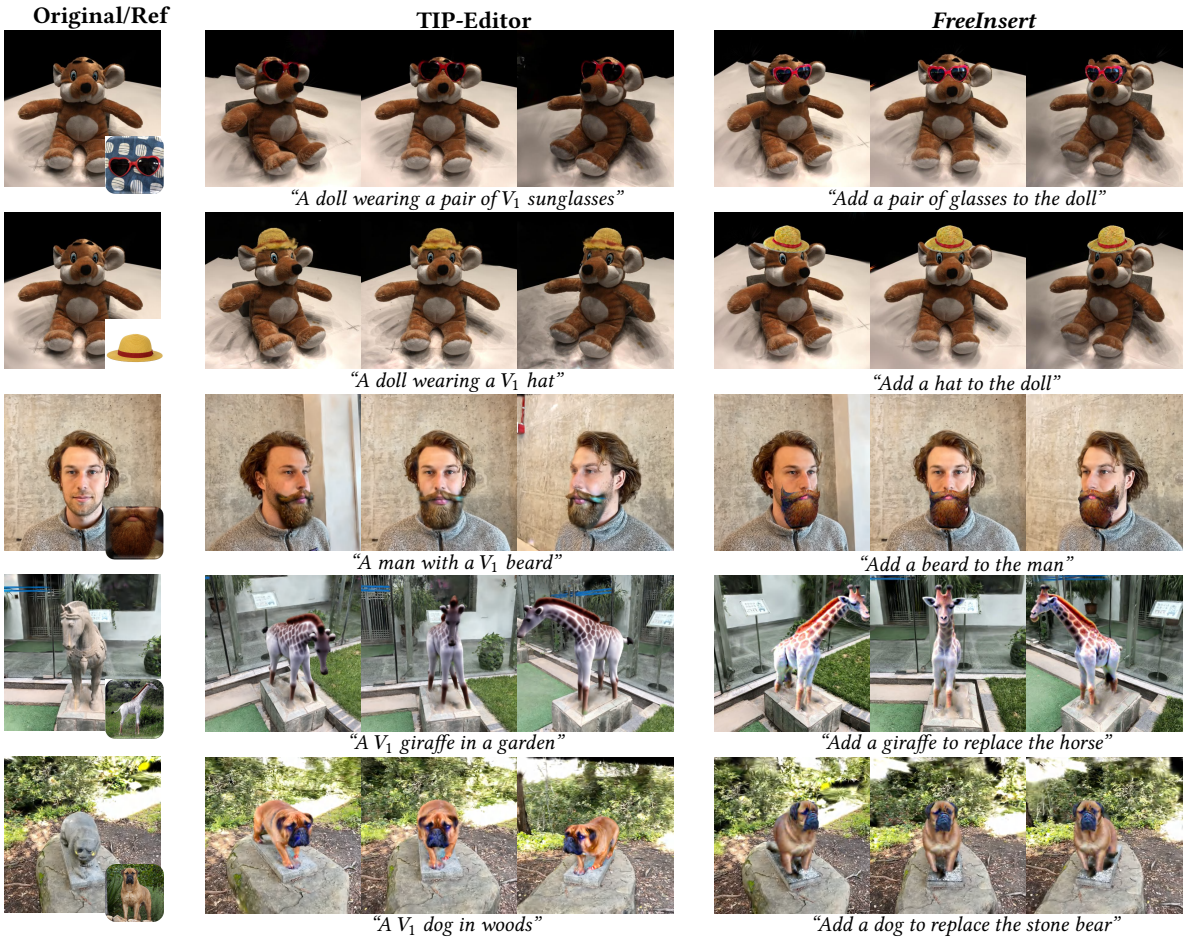

Figure 4: Visual comparisons with TIP-Editor using text-image prompts. Our method achieves competitive results with TIP-Editor, *without relying on 3D bounding boxes*. TIP-Editor struggles to maintain the 3D consistency of the inserted object (e.g., the misaligned hat across views in row 2, column 2, and the right front paw intersecting with the left in row 5, column 2), as its 2D editing process lacks cross-view constraints. Our method produces clearly more 3D-consistent results and more closely resembles the reference image.

to the approach that specifies object appearances (TIP-Editor), our method exhibits higher DINO similarity to the image prompt. The User$_{vote}$ ratings clearly demonstrate that users prefer our method over the baselines. *FreeInsert* is at least an hour faster than TIP-Editor, requires no manual priors, and outperforms faster methods.

Table 2: Quantitative comparisons to SOTA. CLIP$_{dir}$ denotes the CLIP Text-Image directional similarity. DINO$_{sim}$ is the DINO similarity.

Method	CLIP$_{dir}$↑	DINO$_{sim}$↑	User$_{vote}$↑	Time$_{cost}$↓
InstructN2N(GS) [35]	26.76%	-	18.3	**15 mins**
GaussianEditor [6]	27.36%	-	24.7	20 mins
GaussCtrl [41]	25.39%	-	26.3	**15 mins**
TIP-Editor [48]	**30.01%**	83.30%	32.3	2.5 h
FreeInsert	29.48%	**83.45%**	**36.9**	1.1 h

4.3 Ablation Studies

Ablation Visualization Across Stages. To better demonstrate how each stage in *FreeInsert* contributes to the final outcome, we visualize the intermediate results at each step, as shown in Figure 5. Col 2 highlights the attachment region Gaussians within the scene, marked in red. Col 3 shows the initial degrees of freedom (DoF), which are typically sub-optimal. After SSDS refinement, the object achieves a more accurate DoF (Col 4). Finally, Col 5 demonstrates the enhanced object appearance.

Effectiveness of Global-Local Collaborative Spatial Awareness. To verify the effectiveness of global-local collaborative spatial-aware strategy, we conduct ablation studies comparing the following variants: no ssds, only $\mathcal{L}_{ssds\text{-}global}$, only $\mathcal{L}_{ssds\text{-}local}$, and the combination of both. As shown in Figure 6, global prompt like "A man with moustache" often lead to ambiguous placements, while local prompt such as "A moustache is under the nose and above the upper lip" provide precise constraints but may ignore global plausibility. Our method balances both by reweighting key spatial terms and progressively shifting from local to global focus, resulting in more accurate and semantically coherent placements.

Effectiveness of Attention-based localization. We evaluate the contribution of $\mathcal{L}_{loc}$ to enhancing spatial awareness, particularly

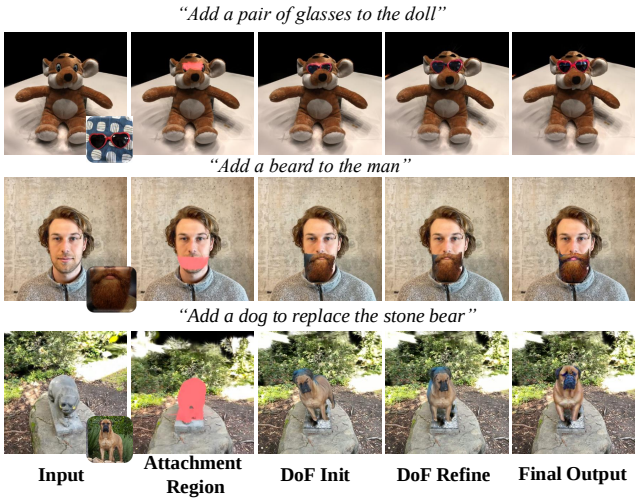

Figure 5: Visualization of different stages in *FreeInsert*.

Figure 6: Ablation of Global-Local Collaborative Spatial Awareness.

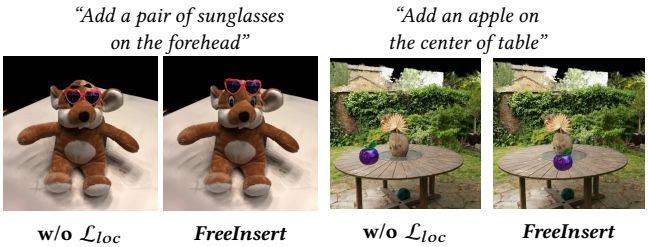

Figure 7: Ablation of Attention-based Localization

for rare or ambiguous placements. As shown in Figure 7, $\mathcal{L}_{loc}$ encourages object position within the bounding box region inferred by the large model, while allowing flexibility to adjust the DoF. Unlike fixed constraints, it softly guides attention toward intended regions, mitigating semantic control failures caused by training data biases.

Comparison of DoF learning directly by Different Multi-Modal Large Language Models vs. *FreeInsert*. To evaluate our DoF learning method, we compare it with state-of-the-art multi-modal LLMs, including GPT-4V [1], Molmo-7B [8], and GPT-o1 [16], which directly predict object DoFs. Translation and scale are derived from multi-view prompts (e.g., "*Point the four coordinates of a bounding box to add [object] to/on [target]*") and lifted to 3D, while rotation follows our initialization. As shown in Figure 8 ("*Add a pair of glasses to the doll*"), our predictions are more plausible. Quantitatively, our method achieves the highest alignment with human preferences in projected mIoU across all cases Table 3.

Effectiveness of Viewpoint Frequency Balancing. Figure 9 compares object appearance optimization when fine-tuning LoRA with a single object image versus combining it with multi-view images

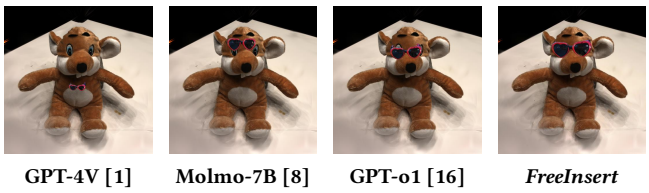

Figure 8: Qualitative comparison of different MLLMs for DoF learning on "Add a pair of sunglasses to the doll".

Table 3: Quantitative analysis of DoF Optimization in FreeInsert.

Metric	GPT-4V	Molmo-7B	GPT-o1	*FreeInsert*
mIoU over 15 cases (%)	68.2	74.7	78.9	89.5

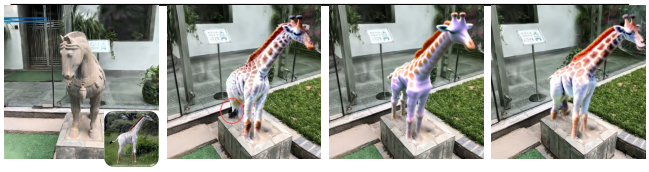

Figure 9: Ablation of Viewpoint Frequency Balancing

at different sampling frequency ratios (*FR*). Using only inserted-object image leads to shape inconsistencies and artifacts across views (item (b)), while incorporating multi-view data improves consistency but may reduce detail. Our experiments show that $FR = 3$ offers the best trade-off between multi-view consistency and single-view image quality.

5 Conclusion and Limitations

In this work, we presented *FreeInsert*, a novel framework for text-driven object insertion in 3D scenes that eliminates the need for spatial priors such as 2D masks or 3D bounding boxes. By disentangling object generation from spatial placement, *FreeInsert* enables unsupervised and semantically guided editing through natural language. Leveraging the reasoning capabilities of foundation models, our method extracts structured semantics from user instructions to guide 3D reconstruction and spatial integration, achieving accurate placement and high visual fidelity. Extensive experiments confirm the effectiveness of our approach in enabling precise, and user-friendly 3D object insertions, paving the way for more scalable and intuitive scene editing in open-world scenarios.

While promising, *FreeInsert* still faces some limitations. It may fail when the underlying 3D reconstruction suffers from severe geometric inconsistencies, such as duplicated limbs or the Janus problem, which cannot be fully compensated by our object-specific refinement. Complex spatial instructions that require hierarchical or relational reasoning (e.g., "*Add ..... to the second layer from the top of the shelf*") may also exceed the capacity of current MLLMs. These challenges are expected to diminish as foundation models for 3D reconstruction and multi-modal reasoning continue to advance. Additionally, in replacement tasks, mismatched contact regions or imprecise 3D bounding boxes can introduce artifacts, which can be alleviated by integrating instance segmentation and local geometry refinement or inpainting.

6 Acknowledgement

This work has been partially supported by the European Union's Horizon Europe research and innovation program under grant agreement No. 101120237 (ELIAS). Bruno Lepri and Nicu Sebe also acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU. This work was also supported by the Tianjin Natural Science Foundation, Key Project, under Grant No. 22JCZDJC00220, "Ultrasound Imaging Algorithm Research for HIFU Thermal Therapy Monitoring" (2022.10–2025.9).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Amir Barda, Matheus Gadelha, Vladimir G Kim, Noam Aigerman, Amit H Bermano, and Thibault Groueix. 2025. Instant3dit: Multiview Inpainting for Fast Editing of 3D Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[3] Alexey Bokhovkin, Quan Meng, Shubham Tulsiani, and Angela Dai. 2025. SceneFactor: Factored Latent 3D Diffusion for Controllable 3D Scene Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 18392–18402.

[5] Chenjie Cao, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. [n. d.]. MVInpainter: Learning Multi-View Consistent Inpainting to Bridge 2D and 3D Editing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[6] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. 2024. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21476–21485.

[7] Yongwei Chen, Tengfei Wang, Tong Wu, Xingang Pan, Kui Jia, and Ziwei Liu. 2024. Comboverse: Compositional 3d assets creation using spatially-aware diffusion guidance. In *European Conference on Computer Vision*. Springer, 128–146.

[8] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. 2024. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146* (2024).

[9] Jiahua Dong and Yu-Xiong Wang. 2023. Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. *Advances in Neural Information Processing Systems* 36 (2023), 61466–61477.

[10] Songxue Gao, Chuanqi Jiao, Ruidong Chen, Weijie Wang, and Weizhi Nie. 2023. Point Cloud Completion Guided by Prior Knowledge via Causal Inference. *arXiv preprint arXiv:2305.17770* (2023).

[11] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[12] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. 2023. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19740–19750.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[14] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. 2023. Lrm: Large reconstruction model for single image to 3d. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[15] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

[16] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720* (2024).

[17] Umar Khalid, Hasan Iqbal, Nazmul Karim, Muhammad Tayyab, Jing Hua, and Chen Chen. 2024. LatentEditor: text driven local editing of 3D scenes. In *European Conference on Computer Vision*. Springer, 364–380.

[18] Subin Kim, Kyungmin Lee, June Suk Choi, Jongheon Jeong, Kihyuk Sohn, and Jinwoo Shin. 2023. Collaborative score distillation for consistent visual editing. *Advances in Neural Information Processing Systems* 36 (2023), 73232–73257.

[19] Juil Koo, Chanho Park, and Minhyuk Sung. 2024. Posterior distillation sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13352–13361.

[20] Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. 2024. Focaldreamer: Text-driven 3d editing via focal-fusion assembly. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 3279–3287.

[21] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).

[22] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. 2024. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9970–9980.

[23] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Marcus A Brubaker, Jonathan Kelly, Alex Levinshtein, Konstantinos G Derpanis, and Igor Gilitschenski. 2025. Watch your steps: Local image and scene editing by text instructions. In *European Conference on Computer Vision*. Springer, 111–129.

[24] Weizhi Nie, Ruidong Chen, Weijie Wang, Bruno Lepri, and Nicu Sebe. 2024. T2TD: Text-3D generation model based on prior knowledge guidance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[25] Weizhi Nie, Weijie Wang, Anan Liu, Jie Nie, and Yuting Su. 2019. HGAN: Holistic generative adversarial networks for two-dimensional image-based three-dimensional object retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15, 4 (2019), 1–24.

[26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research* (2024).

[27] JangHo Park, Gihyun Kwon, and Jong Chul Ye. 2024. ED-NeRF: Efficient Text-Guided Editing of 3D Scene With Latent Space NeRF. In *The Twelfth International Conference on Learning Representations*.

[28] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

[30] Mohamad Shahbazi, Liesbeth Claessens, Michael Niemeyer, Edo Collins, Alessio Tonioni, Luc Van Gool, and Federico Tombari. 2024. InseRF: Text-Driven Generative Object Insertion in Neural 3D Scenes. *arXiv preprint arXiv:2401.05335* (2024).

[31] Ka Chun Shum, Jaeyeon Kim, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. 2024. Language-driven Object Fusion into Neural Radiance Fields with Pose-Conditioned Dataset Updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5176–5187.

[32] Hyeonseop Song, Seokhun Choi, Hoseok Do, Chul Lee, and Taehyeong Kim. 2023. Blending-nerf: Text-driven localized editing in neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*. 14383–14393.

[33] Yanhao Sun, Runze Tian, Xiao Han, XinYao Liu, Yan Zhang, and Kai Xu. 2024. GSEditPro: 3D Gaussian Splatting Editing with Attention-based Progressive Localization. In *Computer Graphics Forum*, Vol. 43. Wiley Online Library, e15215.

[34] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. 2024. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*. Springer, 1–18.

[35] Cyrus Vachha and Ayaan Haque. [n. d.]. Instruct-gs2gs: Editing 3d gaussian splats with instructions (2024). *URL https://instruct-gs2gs. github. io* ([n. d.]).

[36] Dongqing Wang, Tong Zhang, Alaa Abboud, and Sabine Süsstrunk. 2024. Inpaint360: Text-guided 3d-consistent object inpainting on 360-degree neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12677–12686.

[37] Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. 2024. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 20902–20911.

[38] Weijie Wang, Guofeng Mei, Bin Ren, Xiaoshui Huang, Fabio Poiesi, Luc Van Gool, Nicu Sebe, and Bruno Lepri. 2023. Zero-shot point cloud registration. *arXiv preprint arXiv:2312.03032* (2023).

[39] Weijie Wang, Guofeng Mei, Jian Zhang, Nicu Sebe, Bruno Lepri, and Fabio Poiesi. 2025. Fully-Geometric Cross-Attention for Point Cloud Registration. *arXiv preprint arXiv:2502.08285* (2025).

[40] Weijie Wang, Jichao Zhang, Chang Liu, Xia Li, Xingqian Xu, Humphrey Shi, Nicu Sebe, and Bruno Lepri. 2024. UVMap-ID: A Controllable and Personalized UV

Map Generative Model. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 10725–10734.

[41] Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. 2024. Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In *European Conference on Computer Vision*. Springer, 55–71.

[42] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. 2024. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4818–4829.

[43] Yunqiu Xu, Linchao Zhu, and Yi Yang. 2024. GG-Editor: Locally Editing 3D Avatars with Multimodal Large Language Model Guidance. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 10910–10919.

[44] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*. 3836–3847.

[45] Junwei Zhou, Xueting Li, Lu Qi, and Ming-Hsuan Yang. 2025. Layout-your-3D: Controllable and Precise 3D Generation with 2D Blueprint. In *The Thirteenth International Conference on Learning Representations*.

[46] Peng Zhou, Dunbo Cai, Yujian Du, Runqing Zhang, Bingbing Ni, Jie Qin, and Ling Qian. 2024. Edit3D: Elevating 3D Scene Editing with Attention-Driven Multi-Turn Interactivity. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 3401–3410.

[47] Xiaoyu Zhou, Xingjian Ran, Yajiao Xiong, Jinlin He, Zhiwei Lin, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. 2024. GALA3D: Towards Text-to-3D Complex Scene Generation via Layout-guided Generative Gaussian Splatting. In *Forty-first International Conference on Machine Learning*.

[48] Jingyu Zhuang, Di Kang, Yan-Pei Cao, Guanbin Li, Liang Lin, and Ying Shan. 2024. Tip-editor: An accurate 3d editor following both text-prompts and image-prompts. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–12.

[49] Jingyu Zhuang, Chen Wang, Liang Lin, Lingjie Liu, and Guanbin Li. 2023. Dreameditor: Text-driven 3d scene editing with neural fields. In *SIGGRAPH Asia 2023 Conference Papers*. 1–10.