REALM: An MLLM-Agent Framework for Open World 3D Reasoning Segmentation and Editing on Gaussian Splatting

Supplementary Material

1. Details on 3D Editing

We support three 3D editing operations: removal, replacement, and stylization. Given a natural-language query, we utilize our MLLM-Agent framework to identify the desired operation and execute it automatically accordingly.

1.1. Object Removal

The target object can be removed by simply masking the corresponding 3D Gaussians. For large objects that exhibit large-scale occlusion relationships, we can inpaint on 2D images rendered after removing the object using LaMa [4]. The 2D inpainted images can be used as pseudo labels to fine-tune 3D Gaussians.

1.2. Object Replacement

After removing the target object, we insert a TRELLIS [5] -generated Gaussian set $\mathcal{G}_{\text{new}} = \{(\mathbf{x}_j^{\text{new}}, s_j^{\text{new}}, r_j^{\text{new}}, o_j^{\text{new}}, c_j^{\text{new}})\}$ into the scene. To align this set with the scene geometry, we first compute its centroid and recenter all positions:

$$c_{\text{new}} = \frac{1}{|\mathcal{G}_{\text{new}}|} \sum_j \mathbf{x}_j^{\text{new}}, \quad \hat{\mathbf{x}}_j = \mathbf{x}_j^{\text{new}} - c_{\text{new}}, \quad (1)$$

so that $\hat{\mathbf{x}}_j$ denotes each Gaussian's position relative to the model's center. Next, to match the 2D bounding-box width Δu in image space, we compute a uniform scale factor α based on the object's extent along the x-axis:

$$\alpha = \frac{\Delta u / f_x}{\max_j \hat{x}_{j,x} - \min_j \hat{x}_{j,x}}, \quad \tilde{\mathbf{x}}_j = \alpha\, \hat{\mathbf{x}}_j, \quad s'_j = \alpha\, s_j^{\text{new}}. \quad (2)$$

We then apply the canonical orientation D provided by TRELLIS and align to the camera frame via its rotation R_{cam} and translation t_{cam}:

$$R'_j = R_{\text{cam}}\big(D\, R_j^{\text{new}}\big), \quad T_{\text{cam}} = -\, R_{\text{cam}}\, t_{\text{cam}}. \quad (3)$$

Each Gaussian is translated to the insertion point $\mathbf{p}^*$, obtained by back-projecting the 2D box center into 3D:

$$\mathbf{x}'_j = R_{\text{cam}}\, \tilde{\mathbf{x}}_j + T_{\text{cam}} + \mathbf{p}^*, \quad o'_j = o_j^{\text{new}}, \quad c'_j = c_j^{\text{new}}. \quad (4)$$

Finally, the transformed Gaussians are merged into the existing field:

$$\mathcal{G} \leftarrow \mathcal{G} \cup \{(\mathbf{x}'_j, s'_j, r'_j, o'_j, c'_j)\}, \quad (5)$$

[Image Upload]

[System Prompt]

You are a visual reasoning assistant. Your task is to analyze a scene image and identify the key objects present in it. For each object, generate an implicit natural-language query that a human might use to refer to the object indirectly in context. Each query should be rich, imaginative, and avoid explicitly naming or describing the object's category or appearance.

Meanwhile, each object must be assigned a clear and specific object name (e.g., banana, lamp, Gundam figure), which uniquely identifies what the object actually is. These names should be unambiguous and detailed enough for downstream grounding or labeling.

Choose only ONE of the following implicit query styles for the entire image:

(1) location-based (e.g., 'Can you help me find the duck that is on the Rubic\' cube?' or 'Where is the plush toy in the corner of the sofa?'),
(2) attribute-based (e.g., 'Where is the sweet, juicy fruit known for its bright red skin and crisp texture'), or
(3) context-based (e.g., 'There is a child who likes blue. What toy should I find to make her happy?' or 'When you want to watch videos but everyone is asleep, what do you need to use?').

Your output must strictly follow this JSON format:
{ "object_name_1": "implicit query 1", "object_name_2": "implicit query 2", ... }

Object names must be concrete and unambiguous (e.g., banana, not fruit; Game Boy, not device).
Do not use vague categories or general classes as object names.
Queries must not include the object's name or class explicitly.
Do not include any explanations or extra text—output only the JSON dictionary as specified.

[Response]

```json
{
  "spice container": "the thing with a label featuring mushrooms and Chinese characters",
  "green lid spice jar": "the item with a green lid that seems to contain a fine powder",
  "black lid spice jar": "the item with a black lid that appears to hold a different type of spice",
  "plastic container with handle": "the transparent container with a handle, likely used for storing grains"
}
```

Figure S1. **Annotation prompt for REALM3D dataset.** We use MLLM to annotate implicit prompts for REALM3D dataset according to the layout of input images.

1.3. Style Transfer

Given a natural-language editing instruction S, our MLLM-Agent framework automatically generates the style prompt T:

$$T = \text{MLLM}(S). \quad (6)$$

We then isolate the target object using its 3D mask M_i and render each selected view v to obtain the current appear-

Figure S2. **A subset of multi-view images on our REALM3D dataset**

ance I_v and the original unedited image I_v^{orig}. The Instruct-Pix2Pix [1] guidance network produces a stylized reference:

$$I_v^{\text{edit}} \;=\; \text{IP2P}\big(I_v,\; I_v^{\text{orig}},\; T\big). \tag{7}$$

Following InstructNeRF2NeRF [2], the 2D edited images can be lifted to 3D space. We optimize the Gaussian parameters by minimizing the loss

$$\mathcal{L} = \lambda_1 \sum_v \big\| I_v - I_v^{\text{edit}} \big\|_1 \;+\; \lambda_p \sum_v \mathcal{L}_{\text{perc}}\big(I_v,\; I_v^{\text{edit}}\big)\;, \tag{8}$$

where $\mathcal{L}_{\text{perc}}$ is a pretrained perceptual loss. We set $\lambda_1 = 10$, $\lambda_p = 10$, and optimize for 1000 steps using Adam with random view sampling for multi-view consistency. Finally, we render all training cameras to produce the stylized multi-view outputs, transferring the specified style while preserving the object's 3D geometry.

2. Benchmark Preparation

In this section, we describe the detailed preparation process of our benchmark.

2.1. REALM3D Dataset

REALM3D dataset contains more than one hundred 3D scenes (a subset is shown in Fig. S2). We utilize Qwen-2.5-VL to recognize objects and generate corresponding implicit prompts, as shown in Fig. S1.

We apply MLLM to obtain a set of K object labels and corresponding implicit queries:

$$\mathcal{P} \;=\; \big\{(o_k,\; q_k)\big\}_{k=1}^{K} \;=\; \text{MLLM}\big(I\big), \tag{9}$$

Next, for each test view v and each object k, we predict a tight 2D bounding box:

$$b_{v,k} = \text{MLLM}(I_v,\; q_k) \;\in\; \mathbb{R}^4, \tag{10}$$

where $b_{v,k} \;=\; [x_1, y_1, x_2, y_2]$. We then feed this box into SAM to obtain a pixel-precise mask:

$$M_{v,k} = \text{SAM}(I_v,\; b_{v,k}) \;\in\; \{0,1\}^{H\times W}. \tag{11}$$

Through this procedure, we generate 1k+ prompt–mask pairs; any unreasonable prompts are manually filtered to ensure the dataset's quality. We present results in Fig. S5

[Image Upload]

[System Prompt]

You are a visual reasoning assistant. Your input consists of a scene image and the name of a single target object (for example, "apple"). Your task is to produce one implicit natural-language description for that object, choosing the style that best fits the image context:

(1) location-based (e.g., 'Can you help me find the duck that is on the Rubic\' cube?' or 'Where is the plush toy in the corner of the sofa?'),
(2) attribute-based (e.g., 'Where is the sweet, juicy fruit known for its bright red skin and crisp texture'), or
(3) context-based (e.g., 'There is a child who likes blue. What toy should I find to make her happy?' or 'When you want to watch videos but everyone is asleep, what do you need to use?').

Do not include the object's name in the description.
Output exactly one JSON object with a single key–value pair:
{"<object_name>": "<implicit description>"}

[User Prompt]

red apple.

[Answer]

```json
{
"red apple": "The sweet, juicy fruit known for its bright red skin and crisp texture. ”
}
```

Figure S3. **Annotation prompt for LERF and 3D-OVS datasets.** We use LVLM to re-annotate prompts of original LERF and 3D-OVS datasets.

2.2. LERF and 3D-OVS Datasets

GS-Group [6] provides explicit prompt–mask pairs on these datasets. We retain the original masks and re-annotate the objects with implicit prompts. We utilize Qwen-2.5-VL to generate implicit prompts, as shown in Fig. S3.

3. More Results

3.1. Performance on Explicit Queries

We also evaluate REALM and baseline methods using explicit prompt–mask pairs on the 3D-OVS dataset [3]. The evaluation is conducted on the same target objects as presented in the manuscript, but employs the explicit prompts proposed by GS-Group [6]. We present the results in Tab. S1.

Comparing the metrics for implicit and explicit queries reveals that previous methods suffer a pronounced decline in performance with implicit queries. For explicit queries, previous methods find the target object, but sometimes, activate non-target objects (especially for SAM-based methods). In contrast, with LMSeg and GLSpaG, REALM produces more accurate 3D masks, leading to improved evaluation metrics.

Methods	Implicit		Explicit	
	mIoU (%) ↑	mBIoU (%) ↑	mIoU (%) ↑	mBIoU (%) ↑
Gaga	42.53	37.38	73.43	69.87
GAGS	58.46	50.34	84.18	77.37
GS-Group	41.79	38.28	76.31	73.13
REALM (Ours)	**93.68**	**86.02**	**93.62**	**89.71**

Table S1. **Quantitative results for explicit and implicit prompts on 3D-OVS [3] datasets.** The cells with best metrics are highlighted in **bold**.

3.2. Results on Our REALM3D Dataset

We present qualitative comparison results of our REALM3D dataset in Fig. S4.

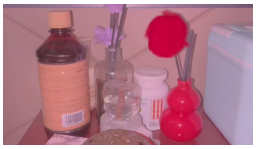
Figure S4. **Qualitative results on our REALM3D dataset.**

100+ Captured 3D Scenes

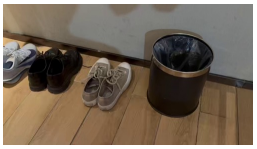
1000+ Annotated Prompt-Mask Pairs


"ink_bottle": "The dark bottle with a black cap sitting on the table.",
"cigarette_packets": "The pink packets stacked neatly on the left side of the table.",
"trash_can": "The light pink trash can with a red lid in the background.",
"box": "The black box partially covered by a wooden board.",
"wooden_board": "The rectangular wooden board leaning against the black box."

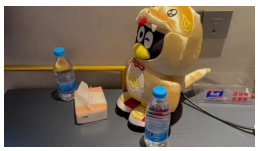

"shampoo_bottle": "The white bottle with a red cap next to the brush.",
"brush": "The blue-handled brush with black bristles on the counter.",
"soap_dispenser": "The tall, gray and silver dispenser on the counter.",
"package": "The yellow package with a tree design behind the shampoo bottle."



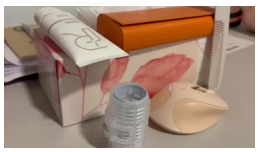

"bottle_with_label": "The tall container with a label and barcode on it.",
"clear_vase_with_flowers": "The transparent vase holding purple flowers and sticks.",
"red_gourd_shape_container": "The red, gourd-shaped container with a fluffy red top."




"shoes_1": "The pair of white sneakers with blue accents.",
"shoes_2": "The black leather shoes with a worn look.",
"shoes_3": "The gray canvas shoes with laces.",
"trash_can": "The dark brown trash can with a metallic rim."




"water_bottle_1": "The clear bottle with a blue cap on the left side of the table.",
"tissue_box": "The orange and white tissue box next to the water bottle.",
"plush_toy": "The large plush toy wearing a yellow hat and outfit, standing on the right side of the table.",
"water_bottle_2": "The clear bottle with a blue cap on the right side of the table."

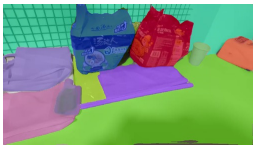



"beauty_product": "The sleek, white tube with a minimalist design and elegant logo.",
"box_with_flowers": "A box adorned with delicate floral patterns in soft pink hues.",
"orange_box": "An orange rectangular box with a smooth finish and a lid.",
"mouse": "A modern, ergonomic mouse with a light beige color and a distinctive shape.",
"hair_curler": "A clear plastic hair curler with multiple sections for styling."


"yellow_cosmetic_bag": "The bright yellow bag with a zipper and handle.",
"clear_bottle_with_white_cap": "The small bottle with a white cap containing a clear liquid.",
"sunscreen_spray": "The tall, blue and orange bottle labeled 'Mistine Aqua'.",
"hair_accessories": "The decorative hairpins and beads on the table."

• • •

• • •

"green_food_package": "The bright green bag with a cartoon character on it",
"purple_food_package": "The purple bag with a yellow character and text",
"white_towel": "The white cloth draped over the edge of the table",
"mask": "The white mask hanging on the left side",
"plastic_bag": "The transparent plastic bag with a green object inside",
"cup": "The white disposable cup near the corner"

Figure S5. **A subset of masks and annotated prompts of REALM3D dataset**

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023. 2

[2] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19740–19750, 2023. 2

[3] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d open-vocabulary segmentation. *Advances in Neural Information Processing Systems*, 36:53433–53456, 2023. 3

[4] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021. 1

[5] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025. 1

[6] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European Conference on Computer Vision*, pages 162–179. Springer, 2024. 3