SIU3R: Simultaneous Scene Understanding and 3D Reconstruction Beyond Feature Alignment

Qi Xu[1,2*], **Dongxu Wei**[2,3*†], **Lingzhe Zhao**[2], **Wenpu Li**[2], **Zhangchi Huang**[2,4], **Shunping Ji**[1†], **Peidong Liu**[2†],
[1]Wuhan University [2]Westlake University
[3]Westlake Institute for Advanced Study [4]Zhejiang University

Project Website: `https://insomniaaac.github.io/siu3r/`

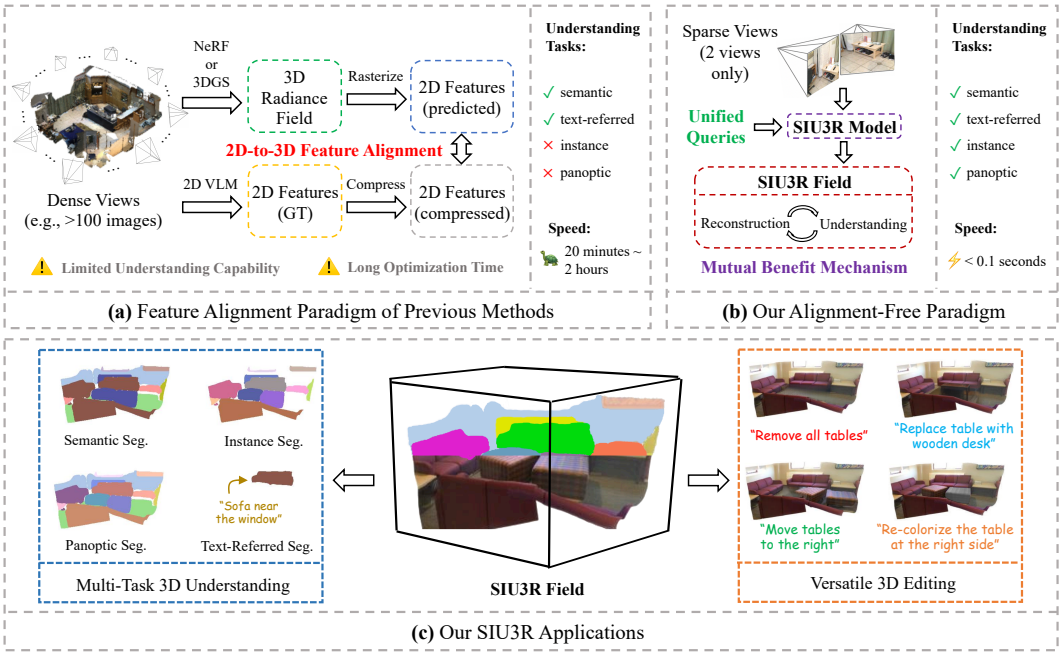


Figure 1: **Simultaneous Scene Understanding and 3D Reconstruction (SIU3R)**. (a) 2D-to-3D Feature alignment paradigm of previous methods. (b) Alignment-free paradigm of our SIU3R method. (c) Versatile 3D reconstruction, understanding (multi-task 3D segmentation) and editing applications of our SIU3R method.

Abstract

Simultaneous understanding and 3D reconstruction plays an important role in developing end-to-end embodied intelligent systems. To achieve this, recent approaches resort to 2D-to-3D feature alignment paradigm, which leads to limited 3D understanding capability and potential semantic information loss. In light of this, we propose SIU3R, the first alignment-free framework for generalizable simultaneous understanding and 3D reconstruction from unposed images. Specifically, SIU3R bridges reconstruction and understanding tasks via pixel-aligned 3D representation, and unifies multiple understanding (segmentation) tasks into a set of unified learnable queries, enabling native 3D understanding without the need of

*Qi Xu and Dongxu Wei contributed equally; † Corresponding author; This work was performed when Qi Xu was an intern at Westlake University.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

alignment with 2D models. To encourage collaboration between the two tasks with shared representation, we further conduct in-depth analyses of their mutual benefits, and propose two lightweight modules to facilitate their interaction. Extensive experiments demonstrate that our method achieves state-of-the-art performance not only on the individual tasks of 3D reconstruction and understanding, but also on the task of simultaneous understanding and 3D reconstruction, highlighting the advantages of our alignment-free framework and the effectiveness of the mutual benefit designs.

1 Introduction

In recent years, 3D scene reconstruction has achieved unprecedented quality and efficiency through differentiable rendering techniques[1, 2], while scene understanding has seen significant advancements via point cloud-based[3–5] or video-based[6–8] approaches powered by large-scale learning. Despite their individual successes, a critical gap remains: current frameworks often treat reconstruction and understanding as separate tasks, hindering the development of end-to-end embodied intelligence systems. In light of this, recent work [9–13] has endeavored to bridge these two tasks for simultaneous understanding and 3D reconstruction within a unified framework.

Current approaches to simultaneous understanding and 3D reconstruction typically follow a 2D-to-3D feature alignment paradigm (Fig.1 a). These methods[9–13] first extract 2D features from pre-trained 2D vision-language models (e.g., CLIP[14], LSeg[15]), then align and fuse them into 3D geometric representations (e.g., neural radiance fields[1] or 3D Gaussians[2]), through a per-scene iterative optimization. The resulting 3D language field jointly encodes scene geometry and semantics, enabling language-guided 3D segmentation via similarity matching with textual features. However, the per-scene optimization leads to heavy expenses for these methods, where each new scene requires dense image captures and hours of training to align 2D features with 3D structures through feature rasterization, severely hindering real-world deployment. The only exception is Large Spatial Model (LSM) [16], a very recent method that employs large reconstruction model and achieves generalizable 2D-to-3D feature alignment by predicting 3D Gaussians and their features in a single forward pass.

However, the aforementioned approaches inherently have the following limitations due to the nature of 2D-to-3D feature alignment. 1) Limited instance-level understanding: Existing feature alignment methods utilize off-the-shelf vision-language models that fall short in identifying instances, resulting in limited capability for instance-level understanding tasks such as instance and panoptic segmentation. 2) Information loss in feature compression: To efficiently embed 2D features into 3D representations and save the memory cost during feature rasterization, existing methods usually need to compress features to lower dimensions (e.g., from 512-dim to 64-dim [16]). Such compression discards fine-grained semantics, degrading performance on tasks requiring precise 3D understanding.

To address the challenges outlined above, we propose **SIU3R**, a novel generalizable framework achieving SIMULTANEOUS UNDERSTANDING and 3D RECONSTRUCTION beyond feature alignment (Fig.1 b). At its core component, SIU3R introduces a Unified Query Decoder and Mutual Benefit Mechanism that bridge 3D reconstruction and scene understanding within a unified SIU3R field through pixel-aligned 3D representation, which enables native 3D understanding through explicit 2D-to-3D lifting rather than implicit 2D-to-3D feature alignment. Specifically, we employ pixel-aligned representation for both reconstruction and understanding, ensuring the subsequently predicted 3D Gaussians and 2D masks can be naturally correlated with each other to enable 3D-level understanding. To ensure mask consistency across different views and understanding tasks, we propose Unified Query Decoder with a single set of learnable queries that shared across different views as well as multiple understanding tasks including semantic, instance, panoptic and text-referred 3D segmentation (Fig.1 c). To further investigate the bidirectional interaction between reconstruction and understanding in the context of their shared representation, we introduce two lightweight modules to encourage mutual benefits between the two tasks, reaching the best of both worlds.

In summary, our main contributions are as follows:

- We propose SIU3R, the first alignment-free framework for generalizable simultaneous understanding and 3D reconstruction, which bridges reconstruction and understanding via pixel-aligned 2D-to-3D lifting, and unifies multiple 3D understanding tasks (i.e., semantic, instance, panoptic and text-referred segmentation) into a set of unified learnable queries. This framework enables native

3D understanding without the need of alignment with 2D models, thereby avoiding limitations on 3D understanding imposed by 2D models and their feature compression.
- To our knowledge, this is the first work that conducts in-depth exploration of inter-task mutual benefits in the realm of simultaneous understanding and 3D reconstruction. To encourage the bidirectional promotion between the two tasks, we incorporate two lightweight modules into our pipeline and achieve significant performance improvements in both tasks.
- Extensive experiments on ScanNet[17] demonstrate that our method achieves state-of-the-art performance not only on the individual tasks of 3D reconstruction and understanding, but also on the integrated task of simultaneous understanding and 3D reconstruction, highlighting the advantages of our alignment-free framework and mutual benefit designs.

2 Related Work

3D Reconstruction. Recent advancements utilizing techniques like Neural Radiance Fields[1, 18–20] and 3D Gaussian Splatting[2, 21–23] have made remarkable progress in both reconstruction quality and rasterization speed. However, they typically require dense image captures as input and time-consuming per-scene optimization. To achieve reconstruction from sparse observations, large reconstruction models have emerged to incorporate 3D representations such as neural radiance fields[24–26], 3D Gaussians[27–33] and 3D point cloud[34, 35] into neural networks, enabling generalizable reconstruction in a feed-forward manner. Among these methods, DUSt3R[34] pioneers pose-free feed-forward 3D reconstruction given only two unposed images, avoiding the obstacles of acquiring camera poses in real world. By combining DUSt3R with additional 3D Gaussian head, [36, 37] can even surpass previous pose-required methods[29, 30] in novel view synthesis. Inspired by this, our method also adopts a pose-free paradigm in 3D reconstruction to ensure generality.

Scene Understanding. Remarkable progress has been witnessed in both 2D and 3D understanding. For 2D understanding, LSeg[15] achieves open-vocabulary semantic segmentation through feature alignment with a vision-language model (e.g., CLIP[14]), which lacks instance-level understanding capabilities such as instance or panoptic segmentation. In parallel, MaskFormer[38] treats semantic segmentation as the task of identifying region proposals of different classes, where a transformer model is used to derive regions from a set of learned class queries. As extensions to MaskFormer, subsequent works further achieve instance-level understanding [39, 40], as well as segmentation on videos[6–8]. Unfortunately, the above 2D methods can only understand scenes from specific camera perspectives, lacking a spatially consistent understanding at the 3D level. For 3D understanding, researchers[3–5, 41–43] typically take 3D point cloud clustered into super points as input and employ proposal-based method, similar to MaskFormer, to segment super points into 3D semantic regions or object instances. However, all of these 3D understanding methods rely on pre-scanned 3D point cloud and pre-processed super points, incurring significant costs in practical applications.

Simultaneous Understanding and 3D Reconstruction. Despite the individual successes of the above methods, they all treat reconstruction and understanding as separate tasks, limiting the potential of building end-to-end embodied intelligence systems. Therefore, recent advances propose to embed language features into neural radiance fields[9, 10, 44–46] or 3D Gaussians[11–13, 47–51] to empower the reconstructed 3D scenes with understanding capabilities. Given dense image captures of a scene, they typically process images into full-image[9–11, 13, 44–46] or object-wise[12, 47–51] features using 2D foundation models (e.g., CLIP[14], LSeg[15]), and then align 3D feature fields with 2D features by imposing losses on the feature rasterization process. Since these methods require time-consuming training for each scene, their efficiency is limited in practical applications. A very recent method Large Spatial Model (LSM)[16] eliminates this drawback by performing feature alignment within a large reconstruction model, achieving generalizable 3D reconstruction and understanding for the first time. Nevertheless, all of these methods follow 2D-to-3D feature alignment paradigm, which leads to limited understanding capability and semantic information loss.

3 Methodology

3.1 Problem Formulation and Pipeline

SIU3R processes sparse unposed multi-view images with corresponding camera intrinsics $\{\boldsymbol{I}^v, \boldsymbol{K}^v\}_{v=1}^V$, where $V \geq 2$ in our setting and denotes the number of input context views, and

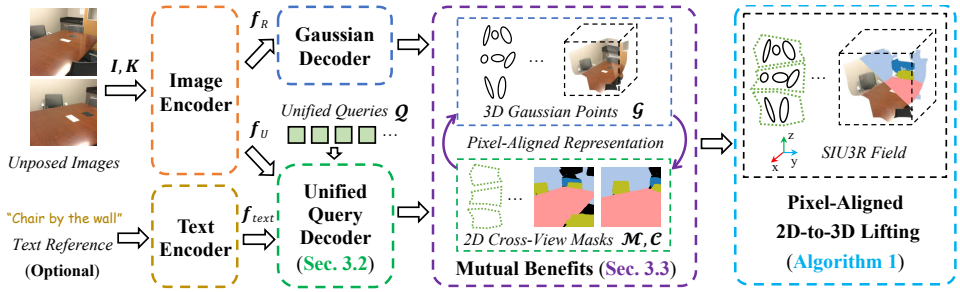


Figure 2: **Pipeline.** Our method consists of Image and Text Encoders for extracting multi-view and text features, Gaussian Decoder for decoding pixel-aligned 3D Gaussians, Unified Query Decoder for decoding pixel-aligned 2D cross-view masks, Mutual Benefit Mechanism for enabling bidirectional promotion between reconstruction and understanding tasks, Pixel-Aligned 2D-to-3D Lifting algorithm for obtaining SIU3R field that enables simultaneous understanding and 3D reconstruction.

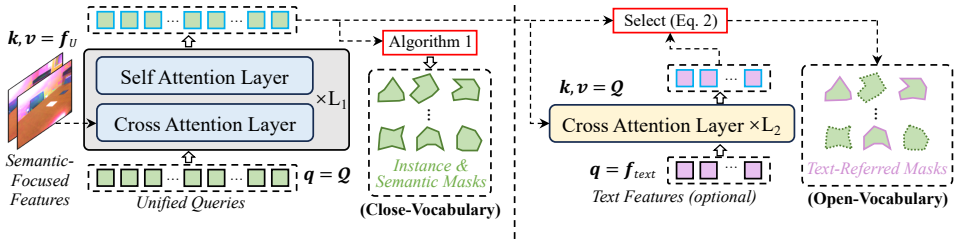
learns a feed-forward network $\mathcal{F}_{\Theta,\mathcal{Q}}$ parameterized by network weights Θ and N_q learnable unified queries $\mathcal{Q} = \{\boldsymbol{q}_n\}_{n=1}^{N_q}$. The network establishes two key outputs: 1) pixel-aligned multi-view 3D Gaussians $\mathcal{G} = \{\boldsymbol{g}_v^{ij}\}_{v,i,j=1}^{V,H,W}$ for 3D reconstruction, where $\boldsymbol{g} = \{\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{c}\}$ is a single gaussian primitive and H, W specify the image resolution; 2) mask prediction logits $\mathcal{M} = \{\boldsymbol{m}_n\}_{n=1}^{N_q}$ where $\boldsymbol{m}_n \in \mathbb{R}^{V\times H\times W}$, and class prediction logits $\mathcal{C} = \{\boldsymbol{c_n}\}_{n=1}^{N_q}$ where $\boldsymbol{c_n} \in \mathbb{R}^{N_c}$ for scene understanding and N_c indicates total classes including background. Formally, the learning objective implements the mapping:

$$\mathcal{F}_{\Theta,\mathcal{Q}} : \{\boldsymbol{I}, \boldsymbol{K}\} \mapsto \{\mathcal{G}, \mathcal{M}, \mathcal{C}\} \tag{1}$$

As illustrated in Fig.2, our pipeline comprises Image Encoder, Text Encoder, Unified Query Decoder (Sec.3.2), Gaussian Decoder, Mutual Benefit Mechanism (Sec.3.3) and Pixel Aligned 2D-to-3D Lifting Algorithm (Algo.1). We design our **Image Encoder** following [52]'s architecture as a Vision Transformer (ViT) enhanced with an adapter module. This configuration enables simultaneous extraction of geometry-focused features $\{\boldsymbol{f}_R^v\}_{v=1}^V$ for reconstruction and semantic-focused features $\{\boldsymbol{f}_U^v\}_{v=1}^V$ for understanding. The extracted semantic features $\{\boldsymbol{f}_U^v\}_{v=1}^V$ along with object queries $\mathcal{Q}$ and optional text features $\boldsymbol{f}_{text}$ extracted by CLIP text encoder[53] are fed into **Unified Query Decoder** to obtain pixel-aligned mask logits $\mathcal{M}$ and class logits $\mathcal{C}$ shared across all views and all understanding tasks, enabling cross-view and cross-task consistent mask predictions. At the same time, **Gaussian Decoder** consisting of a ViT decoder with a DPT head[54] takes geometry features $\{\boldsymbol{f}_R^v\}_{v=1}^V$ as input and predicts pixel-aligned 3D Gaussians $\mathcal{G}$, which are aligned with the 2D cross-view masks. To fully exploit the inherent correlation regarding the shared representation between semantic predictions (i.e., $\mathcal{M}$, $\mathcal{C}$) and geometric predictions (i.e., $\mathcal{G}$), we further propose **Mutual Benefit Mechanism** to encourage collaboration between reconstruction and understanding. In particular, to promote understanding from reconstruction, we propose *Multi-View Mask Aggregation* module, which utilizes 3D geometric clues in $\mathcal{G}$ to aggregate semantic information from all views to improve cross-view consistency of $\mathcal{M}$ and $\mathcal{C}$. Moreover, to improve reconstruction by understanding, we introduce *Mask-Guided Geometry Refinement* module that leverages 2D masks to enforce intra-instance depth continuity for refining reconstructed 3D geometry. Finally, through **Pixel-Aligned 2D-to-3D Lifting**, we can obtain SIU3R field that supports simultaneous understanding and 3D reconstruction.

3.2 Unified Query Decoder

As shown in Fig.3 (a), we employ a set of learnable unified queries $\mathcal{Q}$ to jointly decode cross-view consistent masks for both instance and semantic segmentation tasks, where each query $\boldsymbol{q}_n \in \mathcal{Q}$ explicitly denotes a potential object instance or semantic region that may appear in multiple views. To aggregate semantic logits from multi-view image features, we perform cross-attention between unified queries $\mathcal{Q}$ (query) and semantic-focused multi-view features $\{\boldsymbol{f}_U\}_{v=1}^V$ (key/value), followed by self-attention layer that enables inter-query correlations. This cross-/self-attention block is stacked L_1 times to progressively consolidate semantic information across views, ultimately decoding them into multi-view mask logits $\mathcal{M} = \{\boldsymbol{m}_{n,v}^{ij}\}_{n,v,i,j=1}^{N_q,V,H,W}$ and class logits $\mathcal{C} = \{\boldsymbol{c_n}\}_{n=1}^{N_q}$ through linear projection. Benefiting from the shared representation between $\mathcal{M}$ and reconstructed multi-view 3D Gaussians $\mathcal{G}$, we can correlate each $\boldsymbol{m}_{n,v}^{ij}$ with its 3D Gaussian counterpart $\boldsymbol{g}_v^{ij}$, and lift it to 3D



(a) Cross-View Mask Decoding **(b)** Cross-Modal Matching

Figure 3: **Unified Query Decoder.** In (a), we employ multi-view semantic-focused features $\boldsymbol{f}_U$, unified queries $\boldsymbol{\mathcal{Q}}$, and L_1 stacked cross-/self-attention layer blocks to decode cross-view instance and semantic masks. In (b), we employ L_2 stacked cross-attention layers to select the queries that best match the text features $\boldsymbol{f_{text}}$, and further derive them into text-referred masks.

space for multiple close-vocabulary 3D understanding tasks (i.e., instance, semantic and panoptic segmentation) according to Pixel-Aligned 2D-to-3D Lifting Algorithm derived in Algo.1.

As shown in Fig.3 (b), to enable text features $\boldsymbol{f}_{text} = \{\boldsymbol{f}^t_{text}\}_{t=1}^{N_t}$ as input for open-vocabulary 3D understanding, we further incorporate cross-modal matching module into Unified Query Decoder. Here we assume that our unified queries can perceive all potential object instances and semantic regions. What we need to do is then to identify the queries that best match the text features. Specifically, we employ L_2 cross-attention layers to enable interaction between $\boldsymbol{f}_{text}$ and $\boldsymbol{\mathcal{Q}}$. Then for each text feature $\boldsymbol{f}^t_{text}$, we can match it with the most correlated query $\boldsymbol{q}^t_{text}$, and use its logit prediction to derive text-referred mask, which can also be lifted to 3D for open-vocabulary 3D understanding following Algo.1 same as close-vocabulary tasks. The matching process is derived as follows:

$$\boldsymbol{q}^t_{text} = \arg\max_{\boldsymbol{q}_n \in \boldsymbol{\mathcal{Q}}} \left(\text{Attn}(\boldsymbol{f}^t_{text}, \boldsymbol{\mathcal{Q}}) \cdot \boldsymbol{q}_n\right), \tag{2}$$

where $\text{Attn}(\cdot,\cdot)$ denotes cross-modal attention between text feature $\boldsymbol{f}^t_{text}$ (query) and object queries $\boldsymbol{\mathcal{Q}}$ (key/value) L_2 times, followed by dot product operation with each query $\boldsymbol{q}_n \in \boldsymbol{\mathcal{Q}}$.

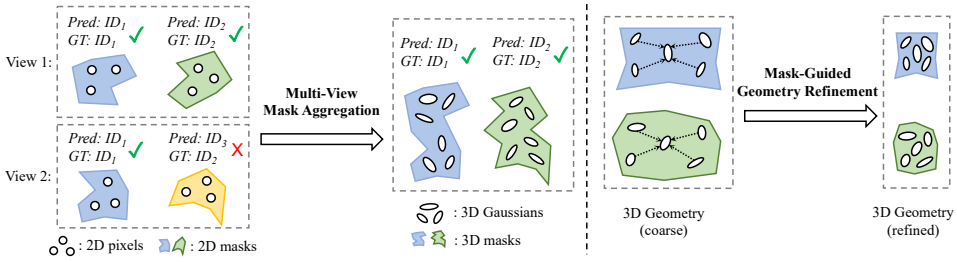
We also introduce the following loss in training to enable matching supervision:

$$\mathcal{L}_{text} = \frac{1}{N_t}\sum_{t=1}^{N_t} \text{CrossEntropy}(\text{Softmax}(\text{Attn}(\boldsymbol{f}^t_{text}, \boldsymbol{\mathcal{Q}}) \cdot \boldsymbol{q}_n), \boldsymbol{\delta}^{gt}_n), \tag{3}$$

where $\boldsymbol{\delta}^{gt}_n$ denotes ground-truth one-hot label that indicates the best matched query. We obtain $\boldsymbol{\delta}^{gt}_n$ by conducting Hungarian matching[39, 55] between $\boldsymbol{\mathcal{M}}$ and ground-truth text-referred masks.

3.3 Mutual Benefit Mechanism

Multi-View Mask Aggregation. Due to potential large viewpoint changes, occlusions, or lighting differences between images from different camera views, the semantic information may vary significantly. Thus, the masks directly predicted by the Unified Query Decoder may exhibit inconsistencies across different views. A typical case is shown in Fig.4 (a), the Unified Query Decoder only considers cross-view semantic correlation of 2D pixels, without realizing that these pixels from different views are probably neighbors in 3D space for the same instance. As a result, the same instance is predicted



(a) R→U: Reconstruction Helps Understanding **(b) U→R**: Understanding Helps Reconstruction

Figure 4: **Mutual Benefit Mechanism.** In (a), our Multi-View Mask Aggregation module utilizes reconstructed 3D Gaussians as geometry clues to improve cross-view mask consistency in 3D understanding. In (b), our Mask-Guided Geometry Refinement module employs segmentation masks as semantic clues to refine geometry of reconstructed 3D Gaussians.

Algorithm 1 *Pixel-aligned 2D-to-3D lifting for simultaneous understanding and 3D recontruction.*

/* Model forward pass */
$\mathcal{G} \leftarrow$ Gaussian Decoder ▷ Pixel-aligned 3D Gaussians
$\mathcal{Q}, \mathcal{M}, \mathcal{C} \leftarrow$ Unified Query Decoder ▷ Last-layer hidden states of unified queries, pixel-aligned multi-view mask logits and class logits
$\boldsymbol{f}_{text} \leftarrow$ CLIP Text Encoder ▷ Obtain textual features of human instructions (**Optional**)
/* Obtain predictions of valid queries, and select the query that matches with the text reference */
$\boldsymbol{kept\_qs} \leftarrow \max(\text{softmax}(\mathcal{C})) > \tau_c \bigcap \arg\max(\text{softmax}(\mathcal{C})) \neq \varnothing$ ▷ Kept queries of high confidences and valid classes
$text\_id \leftarrow \text{Select}(\mathcal{Q}, \boldsymbol{f}_{text})$ ▷ Select the query that best matches the text reference following Eq. 2 (**Optional**)
$\mathcal{Q}', \mathcal{M}', \mathcal{C}' \leftarrow \mathcal{Q}[\boldsymbol{kept\_qs}], \mathcal{M}[\boldsymbol{kept\_qs}], \mathcal{C}[\boldsymbol{kept\_qs}]$; $N'_q \leftarrow \text{len}(\mathcal{Q}')$ ▷ Filter out queries of no interest
$\mathcal{C}', \mathcal{M}' \leftarrow \text{softmax}(\mathcal{C}'), \text{sigmoid}(\mathcal{M}')$ ▷ Turn logits into class confidence scores and multi-view query probability maps
$\mathcal{Z} \leftarrow \mathcal{C}'[\text{None}, :, :, \text{None}, \text{None}] * \mathcal{M}'[:, :, \text{None}, :, :]$ ▷ Calculate class-wise query probability maps $\mathcal{Z} \in \mathbb{R}^{V \times N'_q \times N_c \times H \times W}$
/* Multi-view mask aggregation */
$\mathcal{G}^* \leftarrow \boldsymbol{g}^{*ij}_v; \boldsymbol{g}^* = \{\boldsymbol{\mu}, \boldsymbol{\alpha}, \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{z}\}$ ▷ Replace original 3D Gaussian color attribute $\boldsymbol{c}$ with semantic attribute $\boldsymbol{z}$ by spatial-indexing $\mathcal{Z}$
$\mathcal{Z} \leftarrow \text{rasterize}(\mathcal{G}^*)$ ▷ Fuse multi-view semantics in 3D and propagate them back to the original views via rasterization
/* Derive cross-view 2D masks */
$\boldsymbol{M}_q, \boldsymbol{\mathcal{I}}_q \leftarrow \max(\mathcal{Z}, \text{dim} = 1)$ ▷ Derive class-wise masks and maximum-probability query indices, $\boldsymbol{M}_q, \boldsymbol{\mathcal{I}}_q \in \mathbb{R}^{V \times N_c \times H \times W}$
$\_, \boldsymbol{M}_{sem} \leftarrow \max(\boldsymbol{M}_q, \text{dim} = 1)$ ▷ Derive multi-view semantic masks by maximizing class confidences, $\boldsymbol{M}_{sem} \in \mathbb{R}^{V \times H \times W}$
$\_, \boldsymbol{M}_{ins} \leftarrow \boldsymbol{\mathcal{I}}_q[\boldsymbol{M}_{sem}]$ ▷ Derive multi-view instance masks by taking $\boldsymbol{M}_{sem}$ as class indices, $\boldsymbol{M}_{ins} \in \mathbb{R}^{V \times H \times W}$
/* 2D-to-3D mask lifting based on 3D Gaussians for multi-task 3D understanding */
$\mathcal{G}_{sem} \leftarrow \{\mathcal{G}_{sem\_id} \mid sem\_id \in [1, N_c]\}$; $\mathcal{G}_{sem\_id} \leftarrow \{\boldsymbol{g}^{ij}_v \mid \boldsymbol{M}^{v,ij}_{sem} = sem\_id\}$ ▷ 3D Gaussian-based semantic segmentation
$\mathcal{G}_{ins} \leftarrow \{\mathcal{G}_{ins\_id} \mid ins\_id \in [1, N_q]\}$; $\mathcal{G}_{ins\_id} \leftarrow \{\boldsymbol{g}^{ij}_v \mid \boldsymbol{M}^{v,ij}_{ins} = ins\_id\}$ ▷ 3D Gaussian-based instance segmentation
$\mathcal{G}_{pano} \leftarrow \{\mathcal{G}_{sem}, \mathcal{G}_{ins}\}$ ▷ 3D Gaussian-based panoptic segmentation
$\mathcal{G}_{text} \leftarrow \{\mathcal{G}_{ins\_id} \mid ins\_id = text\_id\}$ ▷ 3D Gaussian-based text-referred segmentation (**Optional**)

with different IDs across different views when semantic information alone is insufficient. Benefiting from the pixel-aligned representation shared between our predicted 2D masks and 3D Gaussians, we can use 3D Gaussians as additional geometric clues to fuse multi-view 2D semantic information in 3D space and propagate it back to the original views to avoid inconsistency. We call this "*Reconstruction Helps Understanding (R→U)*".

Specifically, we propose Multi-View Mask Aggregation module, which first lifts 2D semantic information (i.e., query logits $\mathcal{M}$ and $\mathcal{C}$) from different views to the 3D Gaussians $\mathcal{G}$ for fusion, and then propagate them back to these views through alpha-blending mechanism inherent in the 3D Gaussian rasterization. As highlighted in Algo.1, this module only introduces minor extra computation to the vanilla 2D-to-3D lifting algorithm during the inference phase, without the need for additional training.

Notably, another design choice for the above multi-view aggregation is to apply the 3D Gaussian rasterization earlier to $\boldsymbol{f}_U$ at the stage of feature encoding rather than the mask decoding, which is the core design of previous feature alignment-based methods[11, 12, 16]. However, due to extremely high dimension of 2D features, we have to perform feature compression before rasterization to avoid memory exhaustion, which inevitably corrupts the original semantic information. As demonstrated in Sec.4.3, such early aggregation not only requires additional training but also incurs significant memory overhead, yet its performance is far inferior to our training-free Multi-View Mask Aggregation.

Mask-Guided Geometry Refinement. In general, adjacent 2D pixels within the same object instance or semantic region should correspond to continuous positions in 3D space. Leveraging this prior knowledge, we can use our mask predictions as semantic clues to refine the reconstructed 3D geometries. We call this "*Understanding Helps Reconstruction (U→R)*". As shown in Fig.4 (b), the 3D Gaussians corresponding to adjacent pixels within the same instance may be far apart without refinement, which can lead to unsatisfactory coarse geometry. Thus, to make 3D Gaussians within the same mask to be more clustered, we propose Mask-Guided Geometry Refinement module, which utilizes masks as guidance to enforce intra-instance depth continuity based on the following loss:

$$\mathcal{L}_{cont} = \sum_{k=1}^{K} \sum_{p \in \boldsymbol{M}_k} ||D(p) - \frac{1}{|\mathcal{N}_p|} \sum_{q \in \mathcal{N}_p} D(q)||_2^2, \tag{4}$$

where $\boldsymbol{M}_k$ denotes pixels belonging to the k-th mask, $D(p)$ and $D(q)$ represent the depths at pixel p and one of its neighbored pixel q, and $\mathcal{N}_p$ indicates the neighborhood of p within the same instance.

3.4 Training Objective

Through holistic integration of components, our framework enables end-to-end optimization across the complete learning pipeline. The overall training objective is derived as follows:

$$\mathcal{L} = \lambda_1 ||\boldsymbol{I}(\mathcal{G}) - \hat{\boldsymbol{I}}|| + \lambda_2 \text{LPIPS}(\boldsymbol{I}(\mathcal{G}), \hat{\boldsymbol{I}}) + \lambda_3 \mathcal{L}_{mask} + \lambda_4 \mathcal{L}_{cont} + \lambda_5 \mathcal{L}_{text}, \tag{5}$$

Table 1: **Quantitative comparisons.** "†", "‡" and "⋆" denote reconstruction-only, understanding-only, and simultaneous scene understanding and 3D reconstruction methods, respectively. "-" indicates that the corresponding method do not support the corresponding task. "R→U" and "U→R" denote our Multi-View Mask Aggregation and Mask-Guided Geometry Refinement modules, respectively.

	3D Reconstruction					**Scene Understanding**							
	Depth Estimation		Novel View Synthesis			Context Views (2D-only)				Novel Views (3D-aware)			
	AbsRel↓	RMSE↓	PSNR↑	SSIM↑	LPIPS↓	mIoU_s↑	mAP↑	PQ↑	mIoU_t↑	mIoU_s↑	mAP↑	PQ↑	mIoU_t↑
† pixelSplat[29]	0.1812	0.4106	24.93	0.8065	0.2003	-	-	-	-	-	-		
† MVSplat[30]	0.1697	0.3923	23.80	0.7871	0.2284	-	-	-	-	-	-		
† NoPoSplat[37]	0.0944	0.2434	25.91	0.8147	0.1878	-	-	-	-	-	-		
‡ Mask2former[39]	-	-	-	-	-	0.5466	0.2486	0.6071	-	-	-	-	-
‡ LSeg[15]	-	-	-	-	-	0.2601	-	-	0.2127	-	-	-	-
⋆ LSM[16]	0.07468	0.2190	21.88	0.7336	0.3035	0.2745	-	-	0.1925	0.2707	-	-	0.1905
⋆ **Ours** w/o R→U	**0.07421**	**0.2081**	**25.96**	**0.8220**	**0.1841**	0.5512	0.2529	0.6123	0.4572	-	-	-	-
⋆ **Ours** w/o U→R	0.09619	0.2414	25.51	0.8168	0.1951	0.5893	0.2636	0.6569	0.5125	0.5875	0.2527	0.6456	0.5245
⋆ **Ours**	**0.07421**	**0.2081**	**25.96**	**0.8220**	**0.1841**	**0.5922**	**0.2817**	**0.6612**	**0.5273**	**0.5920**	**0.2714**	**0.6495**	**0.5270**

where I and $\hat{I}$ are rasterized and ground truth images, $\mathcal{L}_{mask}$ is derived from [39, 55], $\mathcal{L}_{mask} = \lambda_{ce}\mathcal{L}_{ce} + \lambda_{dice}\mathcal{L}_{dice} + \lambda_{cls}\mathcal{L}_{cls}$, where $\mathcal{L}_{ce}$ is binary cross-entropy loss, $\mathcal{L}_{dice}$ is dice loss and $\mathcal{L}_{cls}$ is classification loss. We follow [39, 55] set λ_{ce}, λ_{dice} and λ_{cls} to 5.0, 5.0, 2.0. In our training, we leverage both photometric loss and segmentation loss to simultaneously supervise 3D reconstruction and understanding. We set $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ to 1, 0.5, 0.05, 0.05, 1, respectively.

4 Experiments

4.1 Experimental Setup

Implementation Details. We utilize ScanNet[17] for training and validation, the largest public dataset that concurrently provides multi-view images with dense semantic/instance segmentation labels and text-referred segmentation labels[56]. We conduct training on 8 NVIDIA GeForce RTX 4090 GPUs, with our model trained for 100 epochs using a per-GPU batch size of 3 (total batch size of 24) for about 2 hours. AdamW optimizer[57] is employed with an initial learning rate of 1e-4 followed by cosine decay scheduling.

Baselines. Our experiments encompass not only isolated evaluation on separate tasks of 3D reconstruction and understanding, but also integrated evaluation on simultaneous understanding and 3D reconstruction. Therefore, we evaluate our method against three types of baseline methods, all of which are state-of-the-arts on their respective tasks: **1)** Sparse-view 3D reconstruction: pixelSplat[29], MVSplat[30], NoPoSplat[37]; **2)** Scene understanding: Mask2Former[39], LSeg[15]; **3)** Simultaneous scene understanding and 3D reconstruction: LSM[16]. All baseline methods are evaluated on ScanNet dataset under the same protocols as ours for fair comparison. To be more specific, for pixelSplat, MVSplat, NoPoSplat, Mask2Former, and LSM, we re-train their models following the training protocols of their official implementations using the processed ScanNet dataset same as ours. For reconstruction-only methods (i.e., pixelSplat, MVSplat, NoPoSplat), we only use their original rendering losses for supervision. For understanding-only method (i.e., Mask2Former), we only use its mask losses for supervision. For LSeg, since it already possesses general vision-language understanding capabilities, we directly adopt its pre-trained weights for evaluation.

Metrics. For 3D reconstruction, we evaluate the performance from two aspects: depth estimation and novel view synthesis, using depth accuracy metrics (i.e., AbsRel and RMSE) and image quality metrics (i.e., PSNR, SSIM and LPIPS), respectively. For scene understanding, we employ distinct evaluation protocols for 2D-based and 3D-based approaches. Specifically, without reconstructed 3D structures, 2D-based methods can only perform segmentation on the input context views. Therefore, we conduct "2D-only" evaluation on context view segmentation for 2D-based methods. In contrast, 3D-based approaches to simultaneous understanding and 3D reconstruction can perform 3D-level segmentation on the reconstructed 3D structures (e.g., 3D Gaussians in LSM and our method). Thus, for these methods, we leverage their characteristic of 3D-to-2D rasterization to project 3D-level segmentation results onto 2D masks, and conduct "3D-aware" evaluation on novel views. Note that our adoption of such evaluation on novel views is due to the lack of ground-truth segmentation labels for reconstructed 3D structures, which vary across different methods even for the same scene. As for different understanding tasks, we employ mIoU for semantic and text-referred segmentation, mAP for instance segmentation, and PQ for panoptic segmentation, where all metrics are computed with

global IDs as ground truths to penalize inconsistencies across views. To differentiate the metrics for semantic and text-referred tasks, we denote them as "mIoU$_s$" and "mIoU$_t$" in Table 1, respectively.

4.2 Main Results

Quantitative Results. As shown in Table 1, our approach outperforms all baselines across all tasks by a clear margin. For 3D reconstruction, unlike MVSplat and PixelSplat that require camera poses as input, or LSM that relies on ground-truth depth supervision, our framework eliminates dependency on pose and depth priors while achieving superior geometric accuracy and novel view synthesis quality. For scene understanding, existing methods like LSeg and Mask2Former are limited to 2D-only understanding of input context views and specific segmentation tasks. The only method

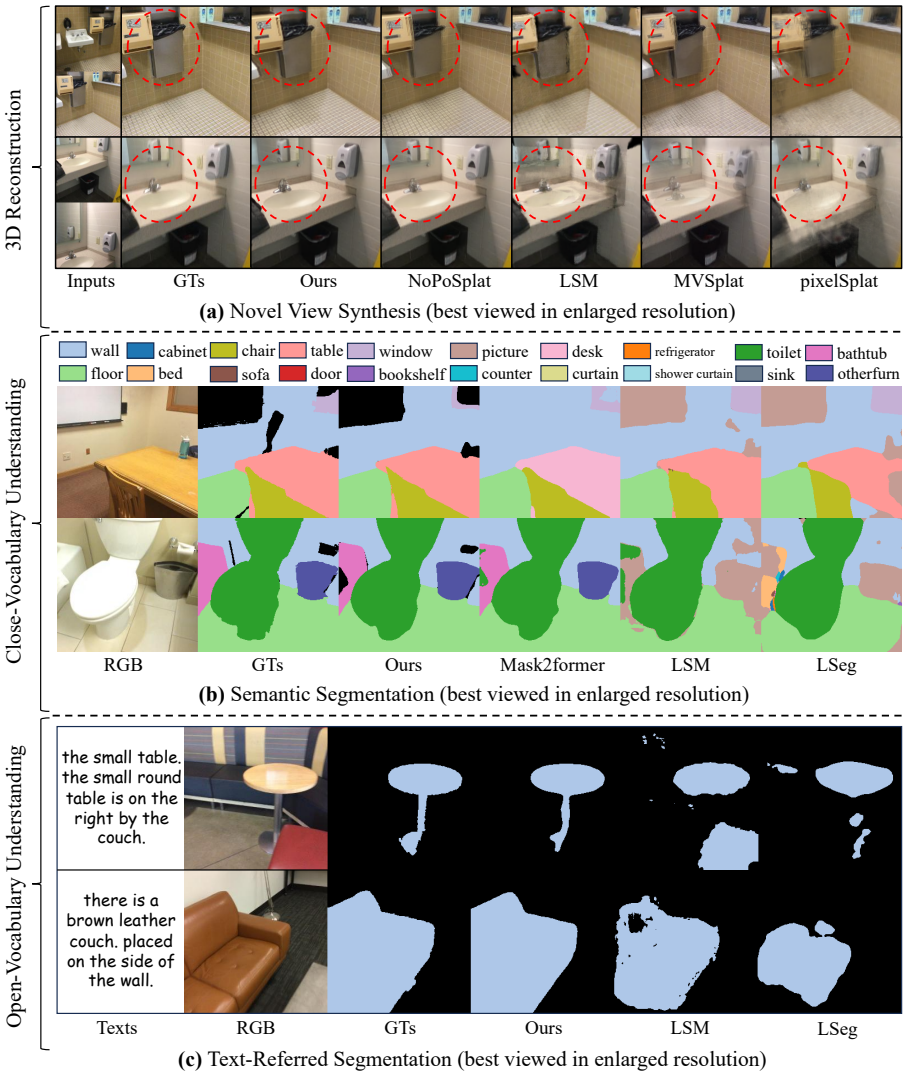


Figure 5: **Qualitative Results.**

that can achieve 3D-aware understanding is LSM. However, its understanding capability is restricted by its source 2D model (LSeg) due to the nature of its feature alignment paradigm. Therefore, LSM can only support semantic and text-referred segmentation same as LSeg. Benefiting from our alignment-free paradigm and simultaneous task modeling, our generalist model supports both 2D and 3D understanding on comprehensive tasks including semantic, instance, panoptic, and text-referred segmentation within a unified framework, significantly exceeding other methods across all metrics. We also conduct experiments to validate the generalizability of our method to more input views, unseen data domains and real-world scenarios. ***Please refer to our appendices*** for more results.

Table 2: **Comparisons between design choices of Multi-View Aggregation.**

	3D Reconstruction			Context Views(2D-Only)				Novel View (3D-aware)				Memory
	PSNR↑	SSIM↑	LPIPS↓	mIoU_s↑	mAP↑	PQ↑	mIoU_t↑	mIoU_s↑	mAP↑	PQ↑	mIoU_t↑	VRAM↓
Early Aggregate w/o train.	**25.96**	**0.8220**	**0.1841**	0.1126	0.051	0.2314	0.0012	0.1015	0.031	0.1765	0.0016	∼**23GB**
Early Aggregate w/ train.	25.62	0.8153	0.1961	0.5442	0.2393	0.6011	0.4025	0.5434	0.2292	0.5849	0.3978	∼33GB
Ours	**25.96**	**0.8220**	**0.1841**	**0.5922**	**0.2817**	**0.6612**	**0.5273**	**0.5920**	**0.2714**	**0.6495**	**0.5270**	∼**23GB**

Qualitative Results. The qualitative results also demonstrate the superiority of our method. As illustrated in Fig.5 (a), thanks to our Mask-Guided Geometry Refinement module that improves reconstructed 3D geometries, our method exhibits better visual quality and fewer artifacts than others in novel view synthesis. As demonstrated in Fig.5 (b), thanks to our simultaneous task modeling and Multi-View Mask Aggregation mechanism, our method can effectively leverage geometric clues to improve understanding, and thus outperform 2D-only methods like Mask2Former with less erroneous inclusions in semantic masks. Besides, since our alignment-free framework decouples us from off-the-shelf 2D vision-language models, the quality of our segmentation masks is much better than methods like LSM and LSeg that rely on language-based querying for segmentation. Similar effects can also be observed in text-referred segmentation results (Fig.5 (c)), where our methods significantly surpasses LSM and LSeg with sharper boundaries and less fragments. ***Please refer to our appendices*** for more results (3D instance and panoptic segmentation, extension to versatile 3D editing, comparisons with more 3D-based baselines, real-world scenarios, etc).

4.3 Ablation Studies

Reconstruction Helps Understanding (R→U). We conduct ablations on our Multi-View Mask Aggregation module (denoted as "R→U" in Table 1). We can see that this module can significantly

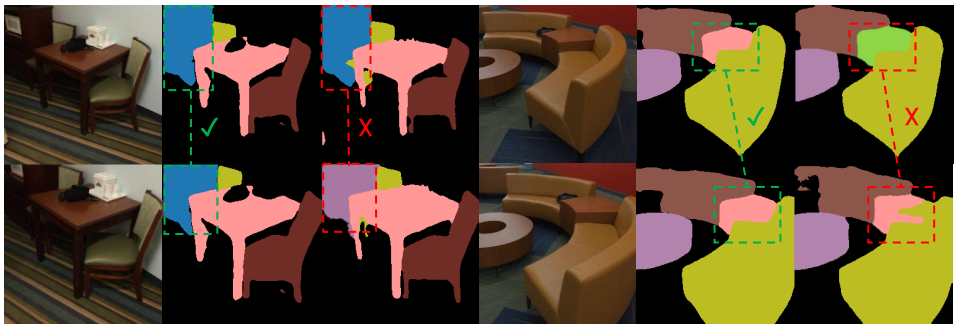


Figure 6: **Ablation on Multi-View Mask Aggregation (R→U).**

improve our performance in both 2D-only and 3D-aware scene understanding, without sacrificing 3D reconstruction accuracy due to its training-free nature. We attribute the improvement to our utilization of reconstructed 3D Gaussians as geometric clues, which can effectively enhance cross-view mask consistency as shown in Fig.6. We also note that, without this module to predict novel-view masks via 3D Gaussian rasterization, we can not enable 3D-aware tasks as denoted by "-" in Table 1.

Understanding Helps Reconstruction (U→R). We also conduct ablations on our Mask-Guided Geometry Refinement module (denoted as "U→R" in Table 1). With this module that employs mask guidance for geometry refinement, we can obtain much better 3D geometries ("depth estimation") and higher visual quality ("novel view synthesis") during reconstruction, which in turn enhances our performance in "Scene Understanding" thanks to our simultaneous modeling of the two tasks.

Design Choices of Multi-View Aggregation. As discussed in Sec.3.3, we can also implement multi-view aggregation via feature rasterization at the early feature encoding stage rather than mask decoding stage. However, as shown in Table 2, such early aggregation leads to poor performance without re-training our model. To incorporate it into training, we have to compress features to lower dimensions to avoid memory exhaustion. Nevertheless, the performance is still far inferior to our original design, while incurs significant training-time memory overhead due to the memory-intensive feature rasterization. We attribute this to the low compatibility between 3D Gaussians and compressed

2D features, which inevitably causes semantic information loss during feature rasterization, and in turn has a negative impact on the intricate forward process of our Unified Query Decoder.

5 Conclusion

We have introduced SIU3R, the first alignment-free framework for generalizable simultaneous understanding and 3D reconstruction. SIU3R unifies multiple understanding tasks into a set of unified learnable queries, which enables native 3D understanding without the need of alignment with 2D models. Two lightweight modules further bring significant mutual benefits between two tasks. Extensive experiments demonstrate the superiority of SIU3R to previous state-of-the-arts on both tasks, as well as its various applications including high fidelity 3D reconstruction, real-time native 3D understanding and versatile 3D editing.

Limitations. Currently, our SIU3R model is only trained on limited data compared to methods such as DUSt3R[34] and VGGT[58], which hinders its generalizability to broader visual domains. Additionally, since all the cross-view mask annotations in ScanNet dataset are obtained by projecting noisy 3D point clouds onto 2D images, the quality of our ground-truth labels is relatively poor, which may have a negative impact on our segmentation accuracy. We hope to address this in the future by introducing higher-quality labels or employing self-supervised methods less dependent on labeled data.

Acknowledgements. This work was supported in part by NSFC under Grant 62202389, in part by a grant from the Westlake University-Muyuan Joint Research Institute, in part by the Westlake Education Foundation, in part by the Headquarters Management Science and Technology Project of State Grid Corporation of China (No. 52090025001L-170-ZN), in part by the Tianjin Science and Technology Plan Project (No. 24YDPYSN00150), and in part by the Open Project Program of the State Key Laboratory of CAD&CG (Grant No. A2513), Zhejiang University.

References

[1] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, August 2020.

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, July 2023.

[3] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE, 2023.

[4] Jiahao Lu, Jiacheng Deng, Chuxin Wang, Jianfeng He, and Tianzhu Zhang. Query Refinement Transformer for 3D Instance Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18516–18526, 2023.

[5] Wei Xu, Chunsheng Shi, Sifan Tu, Xin Zhou, Dingkang Liang, and Xiang Bai. A Unified Framework for 3D Scene Understanding. *Advances in Neural Information Processing Systems*, 37:59468–59490, December 2024.

[6] Tao Zhang, Xingye Tian, Yu Wu, Shunping Ji, Xuebo Wang, Yuan Zhang, and Pengfei Wan. DVIS: Decoupled Video Instance Segmentation Framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1282–1291, 2023.

[7] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[8] Tao Zhang, Xingye Tian, Yikang Zhou, Shunping Ji, Xuebo Wang, Xin Tao, Yuan Zhang, Pengfei Wan, Zhongyuan Wang, and Yu Wu. DVIS++: Improved Decoupled Framework for Universal Video Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–12, 2025.

[9] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in neural information processing systems*, 35:23311–23330, 2022.

[10] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023.

[11] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.

[12] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3D Language Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[13] Xingxing Zuo, Pouya Samangouei, Yunwen Zhou, Yan Di, and Mingyang Li. Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding. *International Journal of Computer Vision*, 133(2):611–627, 2025.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, July 2021.

[15] Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven Semantic Segmentation. In *International Conference on Learning Representations*, October 2021.

[16] Zhiwen Fan, Jian Zhang, Wenyan Cong, Peihao Wang, Renjie Li, Kairun Wen, Shijie Zhou, Achuta Kadambi, Zhangyang Wang, Danfei Xu, Boris Ivanovic, Marco Pavone, and Yue Wang. Large Spatial Model: End-to-end Unposed Images to Semantic 3D. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024.

[17] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, July 2017.

[18] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021.

[19] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023.

[20] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[21] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024.

[22] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[23] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024.

[24] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

[25] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021.

[26] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2021.

[27] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2025.

[28] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024.

[29] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19457–19467, 2024.

[30] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3D Gaussian Splatting from Sparse Multi-view Images. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 370–386, Cham, 2025. Springer Nature Switzerland.

[31] Tianqi Liu, Guangcong Wang, Shoukang Hu, Liao Shen, Xinyi Ye, Yuhang Zang, Zhiguo Cao, Wei Li, and Ziwei Liu. Fast generalizable gaussian splatting reconstruction from multi-view stereo. *arXiv preprint arXiv:2405.12218*, 2024.

[32] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10208–10217, 2024.

[33] Dongxu Wei, Zhiqi Li, and Peidong Liu. Omni-scene: omni-gaussian representation for ego-centric sparse-view scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[34] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[35] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding Image Matching in 3D with MASt3R, June 2024.

[36] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3R: Zero-shot Gaussian Splatting from Uncalibrated Image Pairs, August 2024.

[37] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No Pose, No Problem: Surprisingly Simple 3D Gaussian Splats from Sparse Unposed Images. In *The Thirteenth International Conference on Learning Representations*, October 2024.

[38] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-Pixel Classification is Not All You Need for Semantic Segmentation. In *Advances in Neural Information Processing Systems*, volume 34, pages 17864–17875. Curran Associates, Inc., 2021.

[39] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-Attention Mask Transformer for Universal Image Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022.

[40] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[41] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–824, 2023.

[42] Phuc Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4018–4028, 2024.

[43] Maxim Kolodiazhnyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. Oneformer3d: One transformer for unified point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20943–20953, 2024.

[44] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021.

[45] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Buló, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9043–9052, 2023.

[46] 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.

[47] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Jian Zhang. OpenGaussian: Towards Point-Level 3D Gaussian-based Open Vocabulary Understanding. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024.

[48] Seokhun Choi, Hyeonseop Song, Jaechul Kim, Taehyeong Kim, and Hoseok Do. Click-Gaussian: Interactive Segmentation to Any 3D Gaussians. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 289–305, Cham, 2025. Springer Nature Switzerland.

[49] Guibiao Liao, Jiankun Li, Zhenyu Bao, Xiaoqing Ye, Jingdong Wang, Qing Li, and Kanglin Liu. Clip-gs: Clip-informed gaussian splatting for real-time and view-consistent 3d semantic understanding. *arXiv preprint arXiv:2404.14249*, 2024.

[50] Wenbo Zhang, Lu Zhang, Ping Hu, Liqian Ma, Yunzhi Zhuge, and Huchuan Lu. Bootstraping clustering of gaussians for view-consistent 3d scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 10166–10175, 2025.

[51] Yuzhou Ji, He Zhu, Junshu Tang, Wuyi Liu, Zhizhong Zhang, Xin Tan, and Yuan Xie. Fastlgs: Speeding up language embedded gaussians with feature grid mapping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3922–3930, 2025.

[52] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision Transformer Adapter for Dense Predictions. In *The Eleventh International Conference on Learning Representations*, September 2022.

[53] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2818–2829, 2023.

[54] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision Transformers for Dense Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021.

[55] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing.

[56] Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. ScanRefer: 3D Object Localization in RGB-D Scans Using Natural Language. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 202–221, Cham, 2020. Springer International Publishing.

[57] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[58] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[59] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[60] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*, 2023.

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

[62] 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. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022.

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

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have clearly stated the contributions and scope in the Abstract and Introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We describe the limitations in the Conclusion Section

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We establish our assumptions and proofs upon papers cited.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: : All relevant details are provided within the text of our paper or through the references we have cited.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have included our code and its running instructions in our supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All relevant details are provided within the text of our paper or through the references we have cited.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We conducted ablation experiments, qualitative experiments and quantitative experiments to demonstrate the advantages of the work.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the type of resources used in the experiments section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discusses the broader impact in Conclusion Section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The released model does not have a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the open-source code and datasets mentioned in the paper and adhered to the terms of use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We are committed to releasing our dataset and code as soon as possible.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

In this appendix, we provide additional content to complement the main manuscript:

- Appendix A: Additional Implementation Details
- Appendix B: Comparisons with Per-Scene Optimization Methods
- Appendix C: Extend Our Model to Multi-View Inputs
- Appendix D: Comparison with Other Methods (e.g., DUSt3R, MASt3R, and VGGT)
- Appendix E: Additional Visualizations

A Additional Implementation Details

A.1 Data Preprocessing

As described in Sec.4.1 of our main manuscript, we utilize ScanNet[17] for training and validation. We adopt the official training and validation dataset splitting of ScanNet, and then resize and crop original images to centered images at 256×256 resolution. The camera's intrinsic parameters have also been adjusted accordingly. We followed [37]'s camera conventions, where intrinsics are normalized and extrinsic parameters are OpenCV-style camera-to-world matrices.

Our data samples are obtained by randomly sampling context image pairs with certain overlaps. The overlap is determined by a pair-wise Intersection over Union (IoU) metric as shown in Fig.I. During training, we constrain the IoU to $[0.3, 0.8]$ to randomly select our training samples from scenes. Specifically, the IoU metric can be calculated as follows:

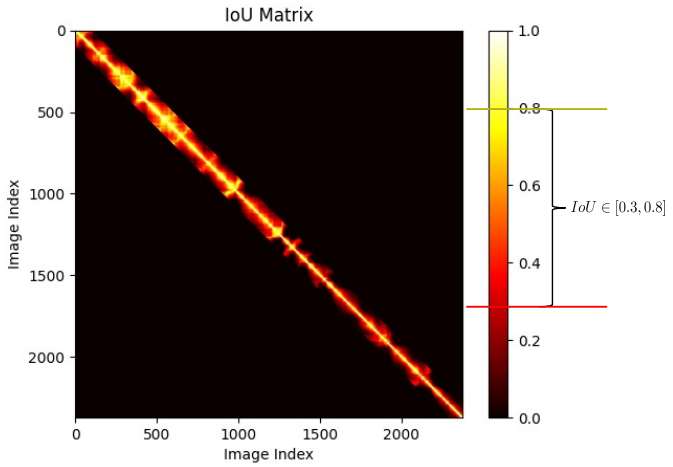


Figure I: **IoU matrix of ScanNet scene0011_00**

1. For a pair of images $\boldsymbol{I}_1, \boldsymbol{I}_2$, obtain their depths $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, poses $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and intrinsic K.
2. Unproject $\boldsymbol{D}_1$ into world coordinates and project them to $\boldsymbol{D}_2$'s camera to obtain $\boldsymbol{D}_1'$. Only depths satisfying $|\boldsymbol{D}_1' - \boldsymbol{D}_2| < 0.1$ are considered as valid.
3. Calculate the intersection over union ratio as:
$$IoU_{i \to j} = \frac{\text{\#valid projected depths}}{\text{\#total depths}}$$
4. Calculate $IoU_{1 \to 2}$ and $IoU_{2 \to 1}$.
5. Define the final IoU as:
$$IoU = \frac{IoU_{1 \to 2} + IoU_{2 \to 1}}{2}$$

The same IoU-based sampling strategy is also adopted in our evaluation, where we select 1,860 context image pairs to formulate the validation set. The curated evaluation benchmark and its processing scripts will be made publicly available for reproducing our results.

A.2 Network Architecture and Hyperparameters

In Table I (a), the order from top to bottom are the network details of Image Encoder, Gaussian Decoder, Unified Query Decoder, respectively. In Table I (b), we specify loss weights for Eq.5 in our main manuscript, which is followed by parameters used in our training phase. To enable the Unified Query Decoder to leverage MASt3R features for scene understanding, we pre-trained the decoder on COCO dataset [59] while keeping the Image Encoder's weights frozen. The pre-trained weights will be publicly released to facilitate further research and development.

(a) Network Architecture		
Image Encoder	architecture	ViT encoder with Adapter[52]
	initialization	MASt3R[35]
	# depth of ViT encoder	24
	# embed dim of ViT encoder	1024
	# attn heads of ViT encoder	16
	positional embedding	RoPE
	# patchsize	16
	# interaction blocks of adapter	[5, 11, 17, 23]
	attention of adapter	MSDeformAttn
	# attention heads of adapter	16
	# inplanes of adapter spatial prior module	64
	# embed dim of adapter spatial prior module	1024
	# ref points	4
	# deform ratio	0.5
Gaussian Decoder	architecture	ViT decoder with DPT head[54]
	initialization	MASt3R[35]
	# depth of ViT decoder	12
	# embed dim of ViT decoder	768
	# attn heads of ViT decoder	12
	# channels of DPT head	83
	# sh degree	4
	# min gaussian scale	0.5
	# max gaussian scale	15.0
Unified Query Decoder	architecture	mask decoder[38, 39]
	# queries	100
	# probability score threshold of queries τ_c	0.5
	# probability score threshold of pixels τ	0.3
	# attn layers for text refer segmentation	6
(b) Hyperparameters		
Loss Weights	# $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$	1.0, 0.5, 0.05, 0.05, 1
Training Details	learning rate scheduler	Cosine
	# epochs	100
	# learning rate	1e-4
	# batch size on each device	3
	training devices	8 * RTX 4090
	optimizer	AdamW[57]
	# beta1, beta2	0.9, 0.95
	# weight decay	0.05
	# warm-up epochs	3
	# gradient clip	1.0

Table I: **Details of network architecture and hyperparameters.** In the table, "#" denotes numerical parameters. We present parameters that specify our network architecture, and parameters used in our loss functions and training phase, in (a) and (b), respectively.

A.3 Implementation Details about Versatile 3D Editing

As shown in Fig.1 of main manuscript, our simultaneous modeling of scene understanding and 3D reconstruction enables diverse 3D scene manipulations through unified pixel-aligned representations, including instance removal, replacement, relocation, and recoloring.

Here we take instance removal as an example to derive the implementation of such 3D editing:

1. Conduct inference to obtain SIU3R field with pixel-aligned 3D masks $\boldsymbol{M}$ and Gaussians $\boldsymbol{\mathcal{G}}$.

2. Remove Gaussians for a specified instance (ID $= ins\_id$):

$$\boldsymbol{\mathcal{G}'} = \boldsymbol{\mathcal{G}} \setminus \{\boldsymbol{g_v^{ij}} | \boldsymbol{M}_{ins}^{v,ij} = ins\_id\}$$

3. The modified Gaussians $\boldsymbol{\mathcal{G}'}$ are rendered into original context views to obtain images $\boldsymbol{\mathcal{I}'}$, with an off-the-shelf diffusion-based inpainting model [60] applied to fill the removed regions while ensuring visual coherence.

4. Conduct inference once again and rebuild SIU3R field from $\boldsymbol{\mathcal{I}'}$.

For other 3D editing tasks (i.e. instance replacement, relocation and recoloring), we adopt a similar approach powered by different diffusion models [61–63].

B Comparisons with Per-Scene Optimization Methods

We also compare our approaches to methods (i.e., Feature-3DGS[11] and NeRF-DFF[9]) that require dense view capturing and per-scene optimization. Both of the two per-scene optimization methods follow a feature alignment paradigm similar to the feed-forward method LSM[16], where their 3D understanding capabilities are powered by off-the-shelf 2D vision language models that can only support language-guided segmentation. To enable the training of Feature-3DGS and NeRF-DFF, we uniformly select dense views (i.e., $\sim$100 images) as input and conduct per-scene optimization for each scene to align 3DGS or NeRF field with 2D features via rasterization. As shown in Table II, our method surpasses all of the feature alignment-based approaches by a large margin in the task of scene understanding, no matter they perform reconstruction in per-scene optimization (Feature-3DGS and NeRF-DFF) or feed-forward (LSM) manner. Besides, benefiting from our align-free framework, our method can further enable instance-level understanding tasks such as instance and panoptic segmentation. Furthermore, our method is the fastest in reconstruction speed, significantly surpassing Feature-3DGS and NeRF-DFF, and leading ahead of LSM. Considering that Feature-3DGS and NeRF-DFF use much more training images than our method, our performance in novel view synthesis is acceptable while achieving the best depth accuracy owing to our mask-guided geometry refinement module. As shown in Fig.III, we also make qualitative comparisons with these feature alignment-based methods, where our method achieves superior mask quality and semantic coherence.

Table II: **Quantitative comparisons with per-scene optimization methods.**

	Depth Estimation		**Novel View Synthesis**			**Scene Understanding**			**Efficiency**
	AbsRel ↓	RMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU ↑	mAP ↑	PQ ↑	Reconstruction Time ↓
Feature-3DGS[11]	0.1546	0.3585	**28.69**	**0.8893**	0.2171	0.3965	-	-	145.30min
NeRF-DFF[9]	0.1846	0.4151	20.12	0.6252	0.5136	0.3410	-	-	2.71min
LSM[16]	0.07468	0.2190	21.88	0.7336	0.3035	0.2745	-	-	0.24s
Ours	**0.07421**	**0.2081**	25.96	0.8220	**0.1841**	**0.5922**	**0.2817**	**0.6612**	**0.13s**

C Extend to Multi-View Inputs

Building upon the insights of some works on feed-forward multi-view reconstruction (e.g., NoPoSplat[37], VGGT[58]), we make minor modifications to our model to support more input views for simultaneous understanding and 3D reconstruction. Specifically, compared to the model with two views, where cross-attention is only performed between the tokens of the two views, for multiple views, we perform cross-attention between the tokens of each view and the concatenated tokens of all other views. This is then followed by our Gaussian decoder and Unified Query Decoder, which predict multi-view 3D Gaussians and masks, respectively. We re-trained four variant models with different number of input views (i.e., 2, 4, 6, 8 views). The quantitative results, shown in the table III, demonstrate that as the number of views increases, our model exhibits improved performance in not only reconstruction but also 3D-aware segmentation on novel views. We think this improvement is primarily due to the availability of additional view information, which reduces geometry uncertainty, and enhances the quality of 3D reconstruction. This, in turn, boosts 3D scene understanding performance, thanks to our mutual-benefit mechanism (i.e., R $\rightarrow$ U).

Table III: **Quantitative results with different numbers of input views.**

#Views	InferTime ↓	InferVRAM ↓	TrainVRAM ↓	AbsRel ↓	RMSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	mIoU ↑	mAP ↑	PQ ↑
2	0.067s	∼ 4G	∼ 24G	0.074	0.208	25.96	0.822	0.184	0.592	0.271	0.650
4	0.097s	∼ 5G	∼ 38.5G	0.072	0.205	26.28	0.822	0.179	0.611	0.283	0.687
6	0.140s	∼ 6G	∼ 56.5G	0.071	0.203	26.45	0.830	0.175	0.617	0.287	0.693
8	0.185s	∼ 7G	∼ 77.5G	0.071	0.201	26.65	0.838	0.170	0.620	0.289	0.697

D Comparison with Other Methods

Since DUSt3R, MAST3R, and VGGT are only designed to reconstruct scenes as 3D points rather than 3D Gaussians / masks, we note that they cannot be evaluated for the tasks of novel view synthesis and multi-task understanding. The comparison results are shown in the table below.

As for reconstruction performance, although DUSt3R, MAST3R and VGGT used much more training data (DUSt3R: 9 datasets, MASt3R: 14 datasets, VGGT: 17 datasets) and ground-truth depths for explicit geometry supervision, our method is only slightly inferior to them in terms of depth accuracy. We think such performance gap is reasonable because our method learns reconstruction solely through the supervision of novel view synthesis (NVS) task and was trained only on a single dataset. Besides, it is worth noting that our method exhibits the best depth accuracy among all the methods that support the novel view synthesis task (NVS), which focuses more on the rendering quality rather than merely on the geometry quality.

As for computational efficiency, our method exhibits faster inference speed compared to DUSt3R, MAST3R, and VGGT. We attribute this to the adoption of complex point post-processing for DUSt3R and MASt3R, and the significantly larger model size for VGGT (ours: 650M, VGGT: 1.1B).

Table IV: **Quantitative comparison on depth estimation task.**

Method	Representation	Task	Runtime ↓	AbsRel ↓	RMSE ↓
DUSt3R[34]	Point	Depth	0.14s	0.058	0.178
MASt3R[35]	Point	Depth	1.09s	0.057	0.175
VGGT[58]	Point	Depth	0.17s	0.052	0.174
MVSplat[30]	3DGS	Depth & NVS	0.06s	0.170	0.392
NoPoSplat[37]	3DGS	Depth & NVS	0.06s	0.094	0.243
LSM[16]	3DGS	Depth & NVS & Segm.	0.19s	0.075	0.219
Ours	3DGS	Depth & NVS & Segm.	0.07s	0.074	0.208

E Additional Visualizations

E.1 Real-World Captured Data

We collect some multi-view images in real-world scenarios using hand-held cellphones to further validate our generalization capability. As shown in Fig.II, we observe that our zero-shot performance in real-world scenes is surprisingly good in both reconstruction and segmentation.

E.2 Instance, Panoptic and Text-Referred Segmentation

In our main manuscript, we have included qualitative comparisons with other methods and demonstrated our superiority in semantic segmentation. Here we provide additional qualitative results of instance and panoptic segmentation for further demonstration. As shown in Fig.IV, compared to 2D-based method Mask2Former[39] that leads to noisy mask boundaries, our method exhibits significantly higher mask quality. As shown in Fig.IV, when performing panoptic segmentation, our method exhibits excellent mask consistency across different views and significantly outperforms other methods in mask quality. Similar effects can also be observed in text-referred segmentation results as shown in Fig.VI. We attribute the superiority of our method to the simultaneous modeling of scene understanding and 3D reconstruction, which effectively leverages 3D geometric clues to aggregate semantic information from different views, and propagates them back to the original views to ensure cross-view consistency and improve segmentation accuracy.

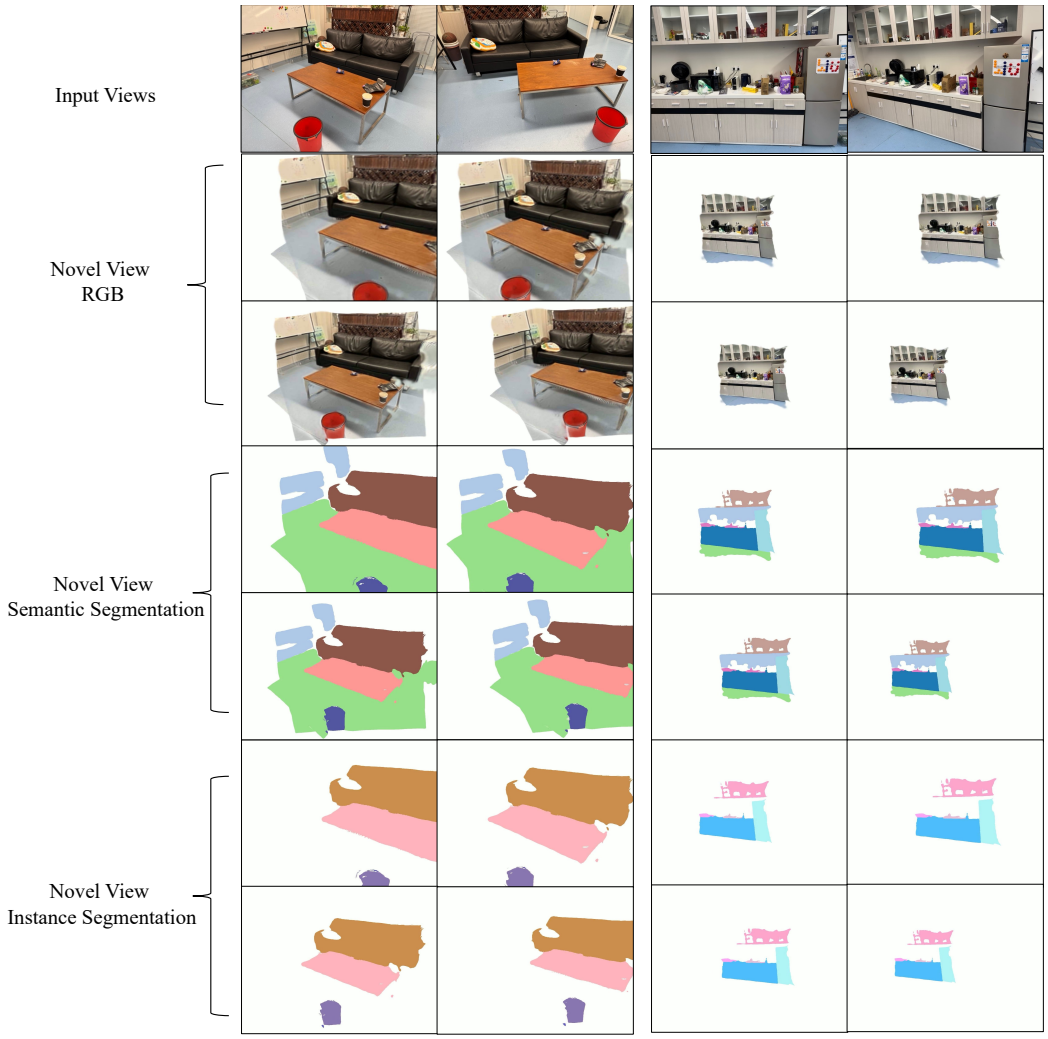


Figure II: **Qualitative result in real-world scenarios.**

E.3 Depth Estimation

As illustrated in Fig.VII, compared to other feed-forward reconstruction methods, our approach achieves significantly superior depth quality with less artifacts and better coherence. We attribute this to our mask-guided geometry refinement module, which ensures geometry consistency within the same object instances under semantic guidance of 2D masks, and thus reduces erroneous depth variations that typically observed in other approaches.

E.4 Versatile 3D Editing

As shown in Fig.VIII, we present a comprehensive set of versatile 3D editing results, demonstrating SIU3R's potential for diverse 3D manipulation applications. Furthermore, this capability establishes an effective baseline that bridges geometric reconstruction, scene understanding and manipulation in 3D environments.

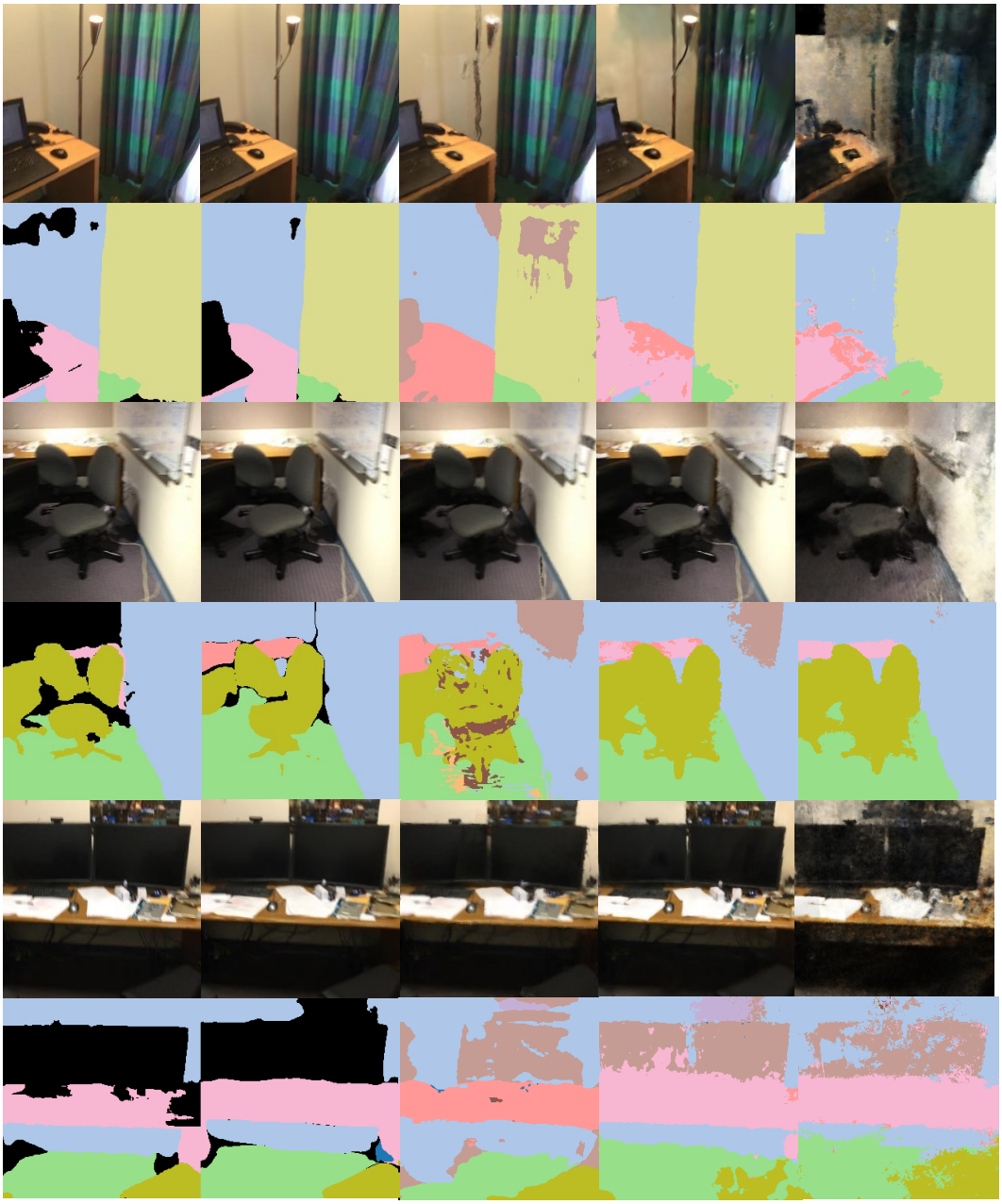


Figure III: **Qualitative comparisons with per-scene optimization methods.**

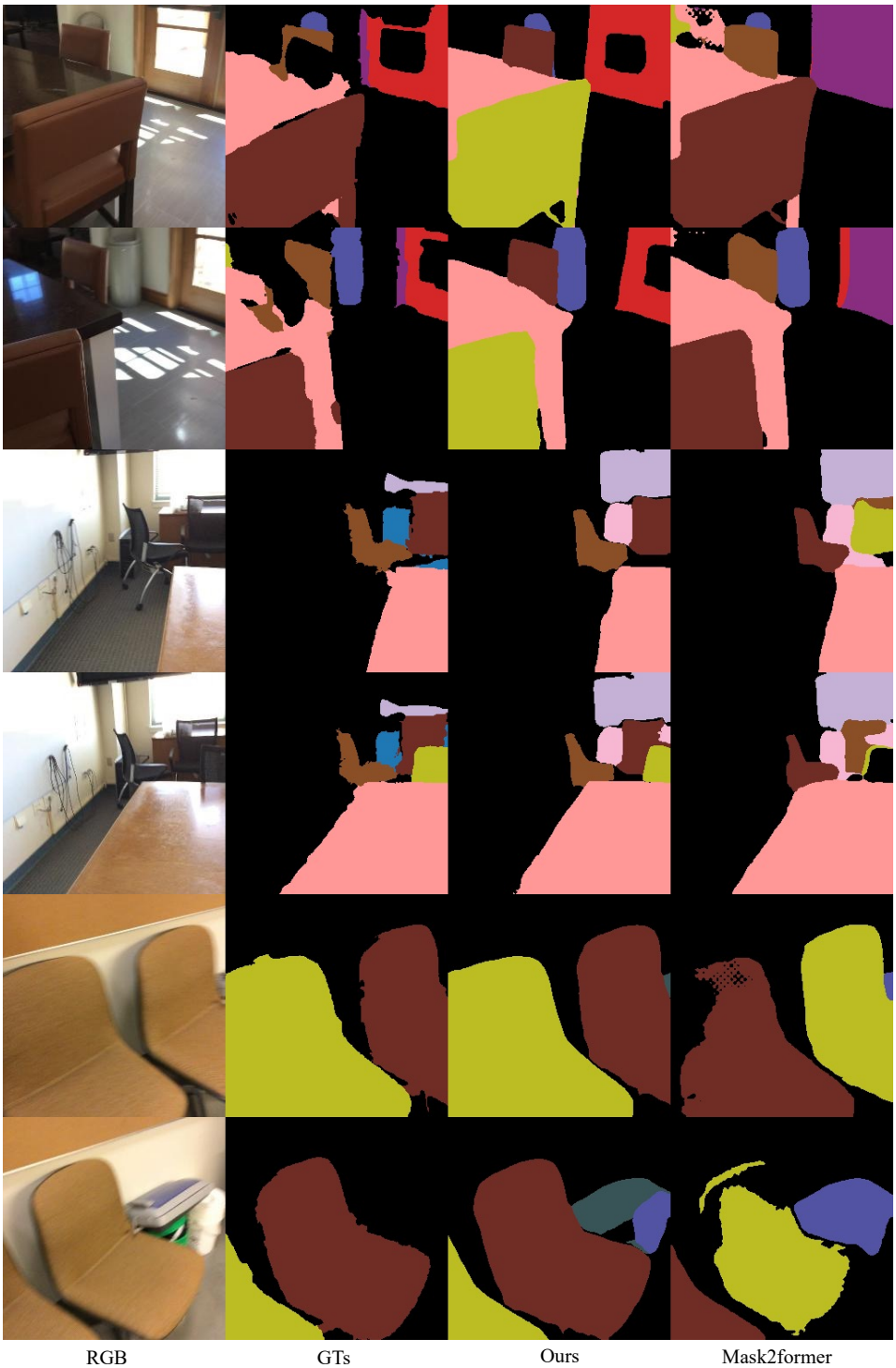


Figure IV: **Qualitative results of instance segmentation.**

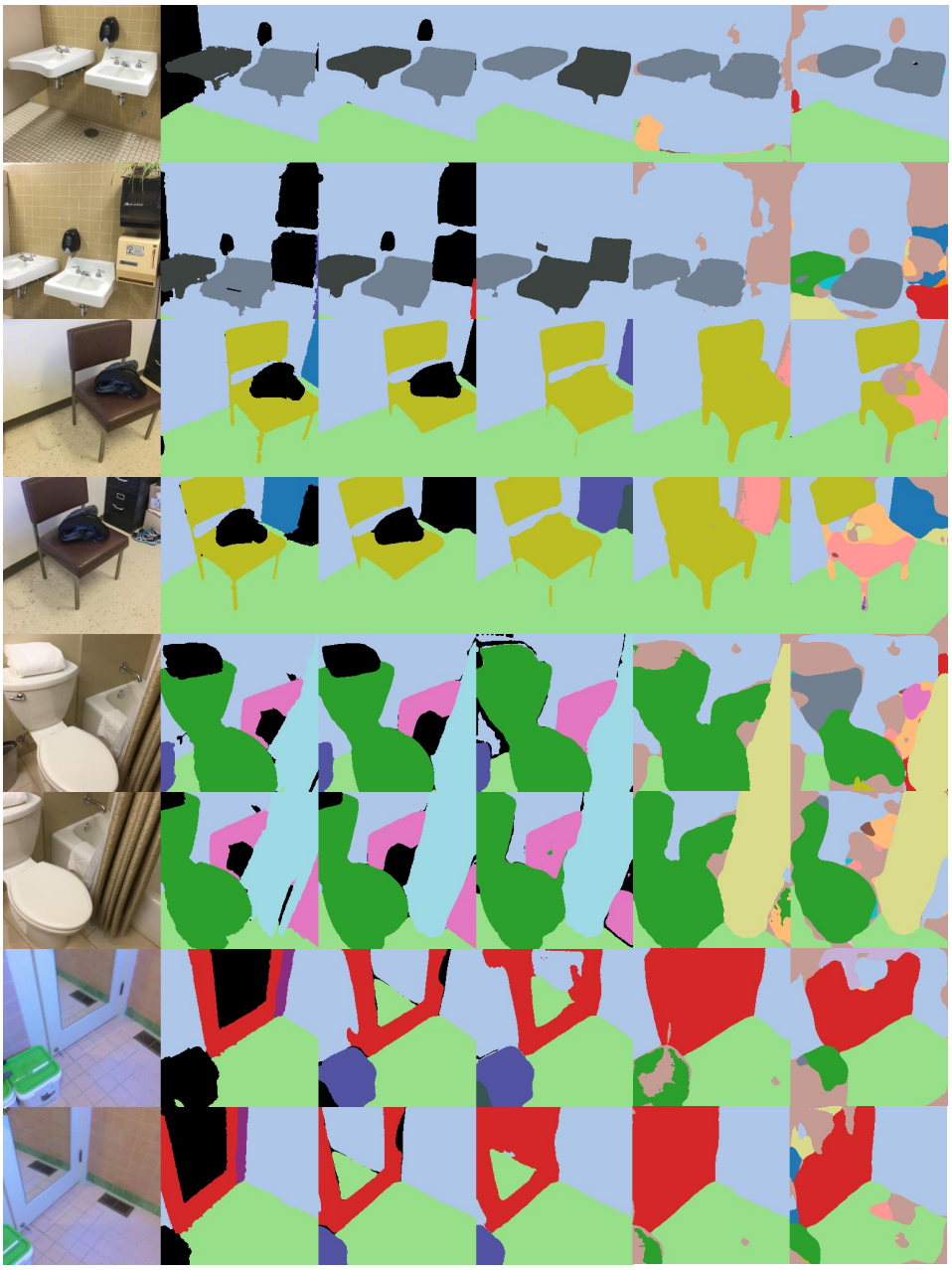


Figure V: **Qualitative Results of Panoptic Segmentation.**

Figure VI: **Qualitative results of text-referred segmentation.**

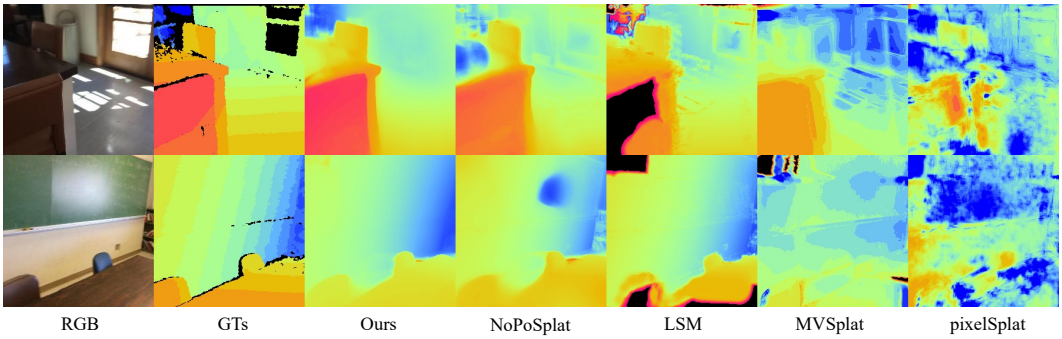


Figure VII: **Qualitative results of depth estimation.**

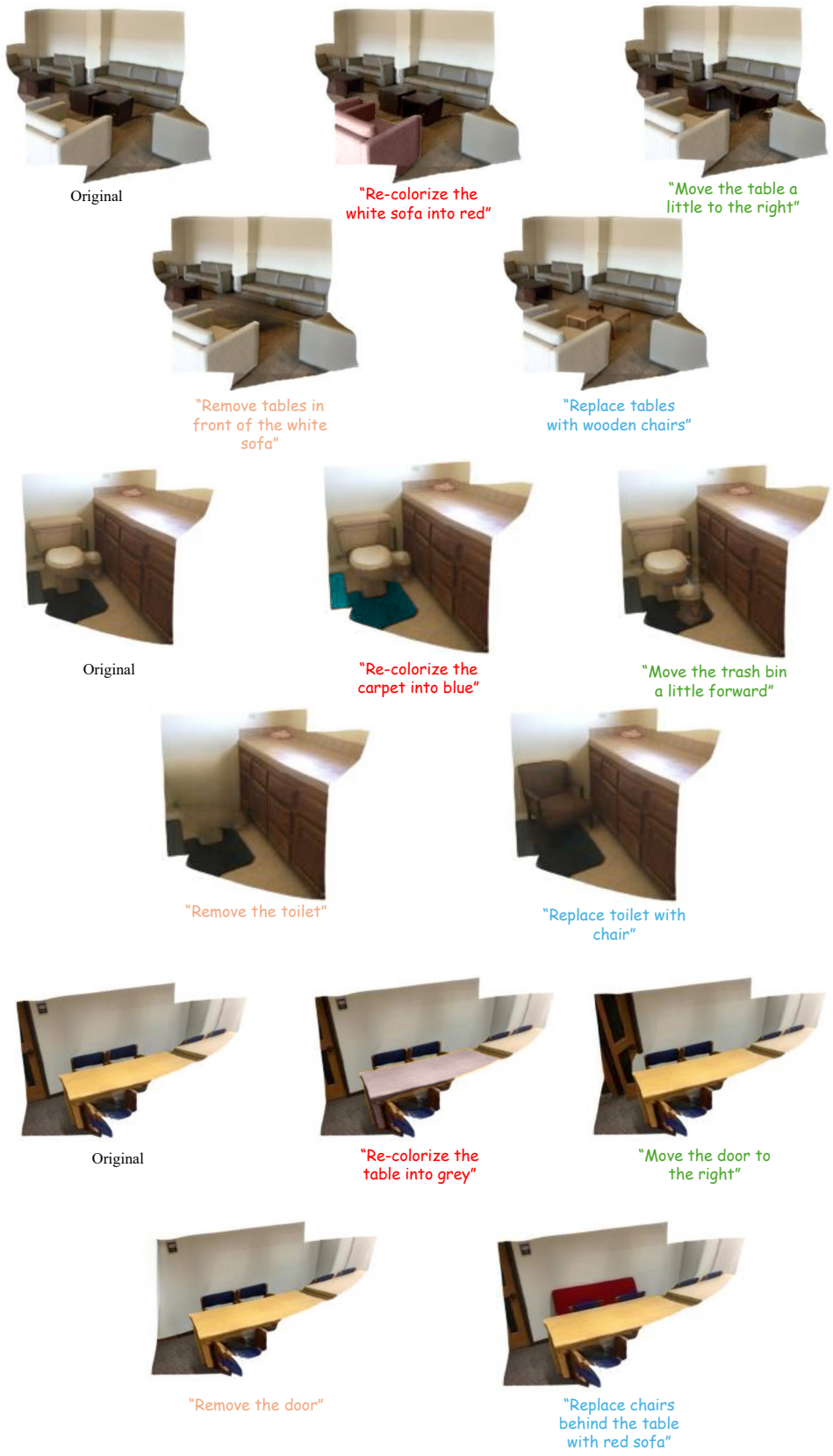


Figure VIII: **Qualitative results of versatile 3D editing.**