

OmniMap: A General Mapping Framework Integrating Optics, Geometry, and Semantics

Yinan Deng, Yufeng Yue*, Jianyu Dou, Jingyu Zhao, Jiahui Wang, Yujie Tang, Yi Yang, and Mengyin Fu

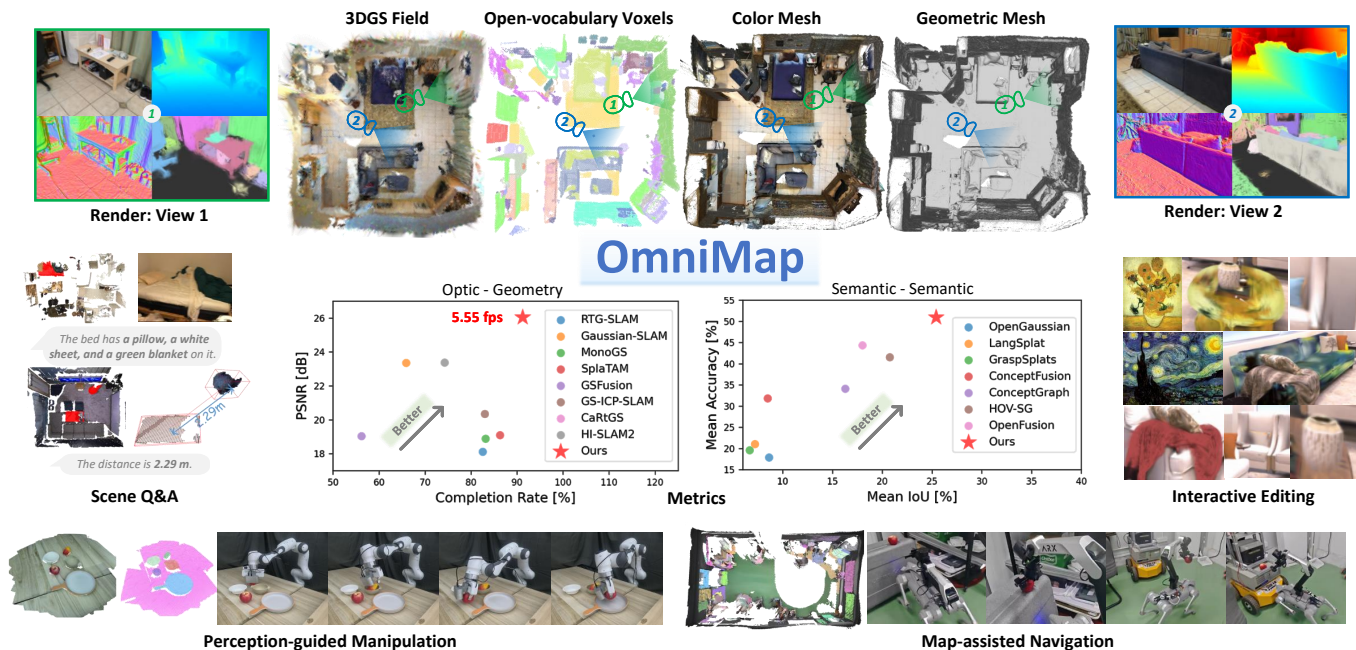


Fig. 1: We introduce **OmniMap**, a general online mapping framework integrating optics, geometry, and semantics. OmniMap incrementally maintains an open-vocabulary instance-level voxel representation and a 3DGS (3D Gaussian Splatting) representation, from which color and geometric meshes are derived. OmniMap supports multi-modal rendering (RGB / depth / normal / instance), and achieves state-of-the-art performance in rendering fidelity, mesh quality, and semantic understanding. This holistic framework enables versatile support for a wide range of downstream applications.

Abstract—Robotic systems demand accurate and comprehensive 3D environment perception, requiring simultaneous capture of photo-realistic appearance (optical), precise layout shape (geometric), and open-vocabulary scene understanding (semantic). Existing methods typically achieve only partial fulfillment of these requirements while exhibiting optical blurring, geometric irregularities, and semantic ambiguities. To address these challenges, we propose **OmniMap**. Overall, **OmniMap** represents the first online mapping framework that simultaneously captures optical, geometric, and semantic scene attributes while maintaining real-

time performance and model compactness. At the architectural level, **OmniMap** employs a tightly coupled 3DGS–Voxel hybrid representation that combines fine-grained modeling with structural stability. At the implementation level, **OmniMap** identifies key challenges across different modalities and introduces several innovations: adaptive camera modeling for motion blur and exposure compensation, hybrid incremental representation with normal constraints, and probabilistic fusion for robust instance-level understanding. Extensive experiments show **OmniMap**’s superior performance in rendering fidelity, geometric accuracy, and zero-shot semantic segmentation compared to state-of-the-art methods across diverse scenes. The framework’s versatility is further evidenced through a variety of downstream applications, including multi-domain scene Q&A, interactive editing, perception-guided manipulation, and map-assisted navigation.

Index Terms—Open-vocabulary, Gaussian Splatting, Mapping, RGB-D perception

I. INTRODUCTION

The quality of a robot’s 3D environmental representation, measured by its accuracy and dimensionality, fundamentally impacts the robot’s task operational performance and execution capabilities. The emergence of Embodied Artificial Intelligence (Embodied AI) has positioned robots as crucial

Manuscript received: 15 May 2025; Accepted 24 August 2025. This article was recommended for publication by Editor Javier Civera upon evaluation of the reviewers’ comments. This work is supported by the National Natural Science Foundation of China under Grant 92370203, 62473050, 62233002, Beijing Natural Science Foundation Undergraduate Research Program QY24180. (Corresponding Author: Yufeng Yue)

Yinan Deng, Yufeng Yue, Jianyu Dou, Jingyu Zhao, Jiahui Wang, Yujie Tang, and Yi Yang are with School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mail: dengyinan@bit.edu.cn; yueyufeng@bit.edu.cn; BruceDou030806@163.com; unique_zhao0210@163.com; wjh@bit.edu.cn; 3120235697@bit.edu.cn; yang_yi@bit.edu.cn).

Mengyin Fu is with the School of Automation, Beijing Institute of Technology, Beijing 100081, China, and the School of Automation, Nanjing University of Science and Technology, Nanjing 210018, China (e-mail: fumy@bit.edu.cn).

The project page of **OmniMap** is available at <https://omni-map.github.io/>.

TABLE I: Comparison of representations of some classic and recent mapping domains.

Robotic Mapping Research Area	Optics High-Fidelity Rendering (✓) Discrete Appearance (✗)	Geometry Detailed Mesh (✓) Sparse Structure (✗)	Semantic Open Understanding (✓) Closed-Set Labels (✗)	Online	Representative Works
(Semantic) Volume Mapping	✗	✗	✗	✓	OctMap [1], Voxblox [2]
Surface Reconstruction	✗	✓	×	✓	KinectFusion [3], BundleFusion [4]
Maps For View Synthesis	✓	✓	×	×	NeRF [5], 3DGS [6]
OV Incremental Mapping	×	✗	✓	✓	ConceptGraphs [7], HOV-SG [8]
OV NeRF/3DGS	✓	✗	✓	×	LERF [9], LangSplat [10]
OmniMap	✓	✓	✓	✓	-

Note: This table outlines a partial yet representative selection of research areas in robotic mapping.

physical agents that bridge AI systems with the real world, driving increasingly sophisticated representation requirements. An ideal environmental representation should enable high-precision, multi-dimensional understanding of the scene, while simultaneously addressing a variety of critical aspects: (1) **optics**: high-fidelity appearance rendering from arbitrary views, (2) **geometry**: fine-grained structural reconstruction, and (3) **semantics**: open-vocabulary object recognition and scene understanding. This representation should support diverse task modes, including interactive virtual operations (scene Q&A and editing) and multi-granular physical interactions (desktop-level manipulation and room-level navigation).

However, current mapping frameworks remain narrowly focused, capturing only limited aspects of the scene. Tab. I summarizes and compares several classic and recent representations in the domain of robotic mapping. Most classic (semantic) volume algorithms rely on fixed-resolution geometry [11]–[13] and closed-set semantic [14]–[16], restricting their functionality to basic tasks like obstacle avoidance and point-to-point navigation. Surface reconstruction methods [3], [4] are capable of recovering fine geometric details, but they typically lack semantic modeling and still fall short in achieving high-quality rendering. While recent advances in Neural Radiance Fields (NeRF) [5] and 3D Gaussian Splatting (3DGS) [6] have demonstrated promising capabilities for photorealistic view synthesis, and have even been adapted into SLAM systems [17]–[19] for online reconstruction, they still primarily address only optical and geometric recovery. Open-vocabulary (OV) mapping leverages the cognitive power of large language models (LLMs) and vision-language models (VLMs) to achieve rich semantic scene interpretation. Yet, existing incremental approaches [7], [8], [20], [21] in this domain often rely on sparse point cloud representations, which inherently lack the density required for high-quality rendering and detailed scene reconstruction. The integration of OV techniques with NeRF/3DGS appears promising for addressing these limitations, however, their batch-style processing makes them challenging to deploy in real-time robotic systems.

This naturally leads us to a fundamental question: *Is it feasible to develop an online mapping algorithm capable of holistically integrating all these scene attributes while simultaneously improving the precision of each individual attribute?* To answer this question, we present OmniMap, as shown in Fig. 1. To better highlight its contributions, we break down this challenge into the following key sub-questions:

1) Representation: how to integrate diverse scene at-

tributes within a unified framework? A key challenge in 3D scene mapping lies in effectively representing and integrating diverse scene attributes within a unified framework. Existing methods often fall short in online robotic systems, being constrained to a single representation [1], [3], [5]–[7] or a naïve combination thereof [22]–[24]. To address this, OmniMap adopts a highly complementary and tightly coupled hybrid 3DGS-Voxel representation. Voxels possess a stable and structured spatial format and often serve as the cornerstone for local probabilistic fusion, though their expressive capacity is inherently constrained by resolution. Differentiable 3DGS provides theoretically arbitrarily fine-grained representations, but due to its positional immobility, it is difficult to directly correlate semantics during optimization, and it faces ambiguity in discriminating newly added Gaussians during incremental mapping. Therefore, OmniMap organically combines the two: voxels act as the fundamental unit for incremental mapping, encapsulating probabilistic semantics within their structure, while newly assigned voxels guide the initialization of Gaussian primitives that underpin precise optical and geometric reconstruction. This integration preserves the fine detail modeling of 3DGS while leveraging the voxel grid’s strengths in probabilistic fusion and structural stability.

2) Optics: how to efficiently handle camera motion blur and exposure inconsistencies?

Camera motion, whether from a mobile chassis or a moving end-effector, frequently induces motion blur in captured images. While some algorithms address this issue through techniques like ray distortion [25], primitive shifting [26], or local linearization [27], their high time complexity makes them unsuitable for online dense mapping systems. Additionally, these methods fail to handle exposure variations, a common challenge in unevenly illuminated environments. In contrast, OmniMap employs a lightweight differentiable camera model, which enhances appearance reconstruction quality while maintaining low computational overhead.

3) Geometry: how to improve geometry stability during 3DGS incremental mapping?

The incremental update of Gaussian primitives poses significant challenges for online 3DGS systems. Current approaches typically follow a brute-force approach: they first back-project RGB-D data to generate colored point clouds, then apply downsampling for Gaussian initialization. This process fails to differentiate observed from unobserved regions, causing redundant primitives in over-scanned areas and insufficient coverage elsewhere. OmniMap addresses these limitations by integrating TSDF-Fusion’s

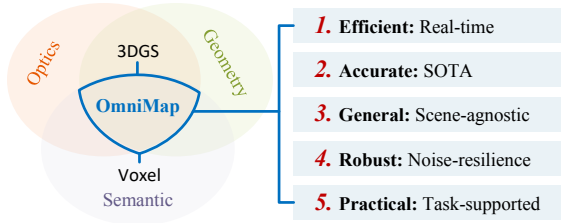


Fig. 2: OmniMap facilitates a holistic representation that integrates optics, geometry, and semantics, and provides five key advantages.

incremental framework (replaceable with other voxel-based mapping methods), strategically initializing Gaussians only within newly assigned voxels to maintain model compactness and geometric stability. Additionally, OmniMap incorporates normal-based supervision directly derived from the rendered depth, significantly enhancing the accuracy of detail modeling and the smoothness of the mesh surface.

4) Semantics: how to robustly recognise and fuse instances in open environments? While VLMs like CLIP enable open-set object recognition within instance masks, their textual inference capabilities remain limited. OmniMap enhances this through an LLM-augmented front-end pipeline. Furthermore, segmentation models often exhibit poor intra-frame accuracy (under/over-segmentation) and inter-frame consistency. Existing incremental methods using IoU-based association with 3D boxes [7] or 2D masks [28] prove sensitive to segmentation noise. OmniMap’s novel solution employs probabilistic instance tuples over TSDF-Fusion voxels, decomposing fusion into two robust subtasks: instance association and live map evolution, modeled as MLE and MAP problems, respectively. This probabilistic formulation naturally preserves uncertainty for improved noise resilience.

In summary, OmniMap successfully integrates three critical scene attributes within a general online mapping framework while achieving significant performance improvements in all dimensions: high-fidelity optical rendering, geometrically stable reconstruction, and open-vocabulary semantic understanding. As demonstrated in Fig. 1, our framework not only surpasses all SOTA methods in quantitative metrics, but also enables versatile downstream applications through its holistic representation. OmniMap offers five key advantages: **a) Efficient:** Real-time operation at sufficient frame rate with compact model size; **b) Accurate:** High precision in rendering, mesh, and cognition; **c) General:** No pre-training or scene-specific optimization required; **d) Robust:** Resilient to sensor and segmentation noise; **e) Practical:** Supports diverse downstream tasks. Our contributions are summarized as follows:

- **Comprehensive Representation:** OmniMap establishes the first general framework bridging optics, geometry, and semantics in online mapping, based on a hybrid 3DGS-Voxel representation.
- **High-Fidelity Optics:** We model the camera motion blurring and exposure process based on four differentiable parameters that efficiently and effectively improve rendering quality.
- **Detailed Geometry:** We propose an incremental 3DGS strategy where newly assigned voxels initialize Gaussian

primitives, and introduce higher-order normal supervision for recovering fine-scale structures.

- **Open Semantics:** We design an efficient instance understanding pipeline and mathematically model the incremental fusion process as two subtasks, enabling adaptation and recovery from noise.
- **SOTA Performance:** OmniMap achieves superior metrics in image rendering, mesh reconstruction, and zero-shot semantic segmentation, while maintaining a high frame rate and compact model size.

II. RELATED WORKS

In this section, we review previous related research on robotic mapping, including discrete reconstruction, 3DGS-based dense mapping, and open-vocabulary representation.

A. Discrete Reconstruction

Classical representation structures for robotic mapping, such as point clouds and voxels, have been extensively developed over the past decades. The widely adopted OctoMap [1] offers an efficient 3D probabilistic volumetric representation via octrees. Meyer-D et al. [29] model dynamic environments by introducing state transition probabilities in Hidden Markov Models for occupancy grids. Beyond occupancy, TSDF and ESDF representations are also widely integrated into online reconstruction frameworks [30]. The advancement of deep learning has significantly accelerated progress in robotic semantic understanding, enabling numerous recent breakthroughs. DA-RNN [31] introduces Data Associated Recurrent Neural Networks for semantic annotation of RGB-D video using a novel recurrent architecture. SemanticFusion [32] integrates CNN-based semantic predictions with ElasticFusion and refines them via CRF-based map updates. Semantic-OcTree [15] proposes a Bayesian multi-class octree mapping method, where semantic categories are probabilistically updated. However, these methods are purely discrete and overlook the inherently continuous nature of the real world.

To construct smoother occupancy maps, many methods relax the assumption of voxel independence. GPmap [33] models point dependencies via nonparametric Bayesian inference, which has also been extended to semantic mapping [34]. Hilbert maps [35] leverage fast kernel approximations for efficient training, enabling real-time incremental 3D mapping. Recently, Bayesian kernel inference with lower computational complexity has gained attention. BGKOcTree [36] applies sparse kernels and nonparametric inference to enhance efficiency, while Lu et al. [37] extends binary occupancy mapping to multiclass and multitask domains. To address over-inflation issues, S-MKI [38] and SEE-CSOM [14] introduce adaptive multi-entropy kernel inference, improving both efficiency and accuracy in continuous semantic occupancy mapping.

Although these frameworks have served as milestones in robotic mapping, their reliance on discrete sparse representations limits scalability. While continuous occupancy mapping alleviates some issues, it remains constrained by map resolution, lacks rendering capabilities, and is restricted to closed-set semantics, falling short of the representation demands in embodied intelligence applications.

B. 3DGS-Based Dense Mapping

In recent years, learning-based implicit mapping methods have emerged, inspired by NeRF’s strengths in continuous representation and appearance modeling. These approaches have been applied to object reconstruction [39]–[41] and incremental SLAM [42]–[44], enabling the generation of photorealistic renderings with compact models. Although some efforts [45]–[47] focus on accelerating vanilla NeRF, the computational burden of volumetric rendering and implicit representation continues to pose a significant challenge for the practical deployment of NeRF-like methods. More recently, 3DGS [6] achieved real-time novel view rendering without sacrificing quality by leveraging a tile-based differentiable splatting technique. However, its original design is tailored for offline processing and typically requires tens of minutes to batch process small-scale datasets, making it unsuitable for exploratory robotics where full-scene sensor data is unavailable upfront.

To address this, recent studies have integrated 3DGS with visual SLAM systems and proposed targeted strategies for online optimization. GS-SLAM [48] employs an adaptive scaling strategy to dynamically add new or remove noisy Gaussians, while selecting Gaussians for camera pose optimization through a coarse-to-fine approach. By decomposing the scene into independently optimized sub-maps, Gaussian-SLAM [17] enables online optimization of newly explored regions regardless of scene scale. SplatTAM [18] leverages rendered depth and silhouette images to effectively capture scene density distribution to guide Gaussian insertion. RTG-SLAM [49] densifies Gaussians based on geometry and optics, enforcing complete opacity or transparency to better represent dominant and residual surface colors, respectively. MonoGS [19] places Gaussians through depth back-projection and applies isotropic regularization to reduce artifacts in novel views. Splat-SLAM [50] dynamically adapts to keyframe pose and depth updates through active map deformation.

Recent approaches have focused on balancing the quality and efficiency of dense 3DGS mapping. GS-ICP-SLAM [51] integrates Generalized Iterative Closest Point with 3DGS and adds mapping keyframes alongside tracking keyframes. CaRtGS [52] uses an adaptive computational alignment strategy that effectively tackles insufficient optimization, long-tail convergence, and weakly constrained densification. HI-SLAM2 [53], built upon DROID-SLAM, incorporates monocular geometry priors (depth and normals) to enhance 3DGS geometric estimation. However, these methods lack explicit 3D Gaussian specification, making them susceptible to over- and under-fitting. Most closely related to our work is GSFusion [22], which combines TSDF-Fusion and 3DGS in a hybrid representation. Yet, it still relies on depth backprojection for Gaussian initialization, using voxels only as a filtering criterion.

More importantly, existing methods often neglect two key challenges introduced by camera motion: motion blur during movement and exposure inconsistency across viewpoints in non-uniformly lit scenes. While offline techniques [26] exist for blur mitigation, their high computational cost hinders real-

time use. OmniMap addresses this by explicitly modeling per-frame camera parameters with four differentiable variables to capture blur and exposure configurations, and experimental results validate the effectiveness of this design.

C. Open-Vocabulary Representation

To extend semantics from closed-set pre-trained labels to open-set categories, researchers have developed open-vocabulary mapping systems leveraging VLMs and LLMs for zero-shot generalization. Early approaches typically extracted pixel-level 2D VLM features, which are then projected or distilled into 3D space. ConceptFusion [20] combines global and local features into pixel-aligned ones, which are subsequently fused into a 3D point cloud via multi-view fusion, showcasing cross-modal inference. Based on NeRF, DFF [54] distills the knowledge of self-supervised 2D image feature extractors such as into a 3D feature field optimized in parallel to the radiance field. Similarly, LERF [9] learns a dense, multi-scale language field inside NeRF by volume rendering CLIP embeddings. OpenObj [55] builds multi-granularity language fields by distilling part features into objects’ NeRFs.

Some other methods enhance the field by leveraging a collection of 3D Gaussians. LangSplat [10] leverages SAM’s multi-granularity masks for feature extraction and CLIP encoding, incorporating an autoencoder to embed compressed semantics into Gaussians. Gaussian Grouping [56] augments each Gaussian with a compact Identity Encoding, enabling grouping of Gaussians based on object instances in the 3D scene. OpenGaussian [57] trains feature consistency within instances and employs a two-phase codebook to distinguish between coarse and fine features. Although integration with 3DGS enables fast feature rendering in these methods, most employ a two-stage optimization process: geometry first, followed by semantics. To date, no approach has directly incorporated open-vocabulary understanding into the incremental mapping of 3DGS, as the continuous displacement of Gaussian centers during optimization disrupts the stability of their associated semantic attributes.

Recently, several algorithms have advanced incremental instance-level or region-level open-vocabulary mapping. ConceptGraphs [7] compute associations between frames according to the IoU of the bounding box of the point cloud obtained by back-projection of the instance segmentation mask, and fuse features from different instances to obtain the final instance-level point cloud. OpenGraph [21] extends ConceptGraphs to a wide range of outdoor environments, and builds hierarchical scene graphs for structured querying. OpenFusion [28] extracts region-level features with SEEM and achieves fast cross-frame correlation through 2D mask and confidence rendering. However, these methods rely on empirically determined IoU threshold parameters and cannot fully mitigate the impact of robust segmentation at the front end. In contrast, OmniMap appends probabilistic instance tuples to each voxel, using probabilistic modeling to improve noise recovery.

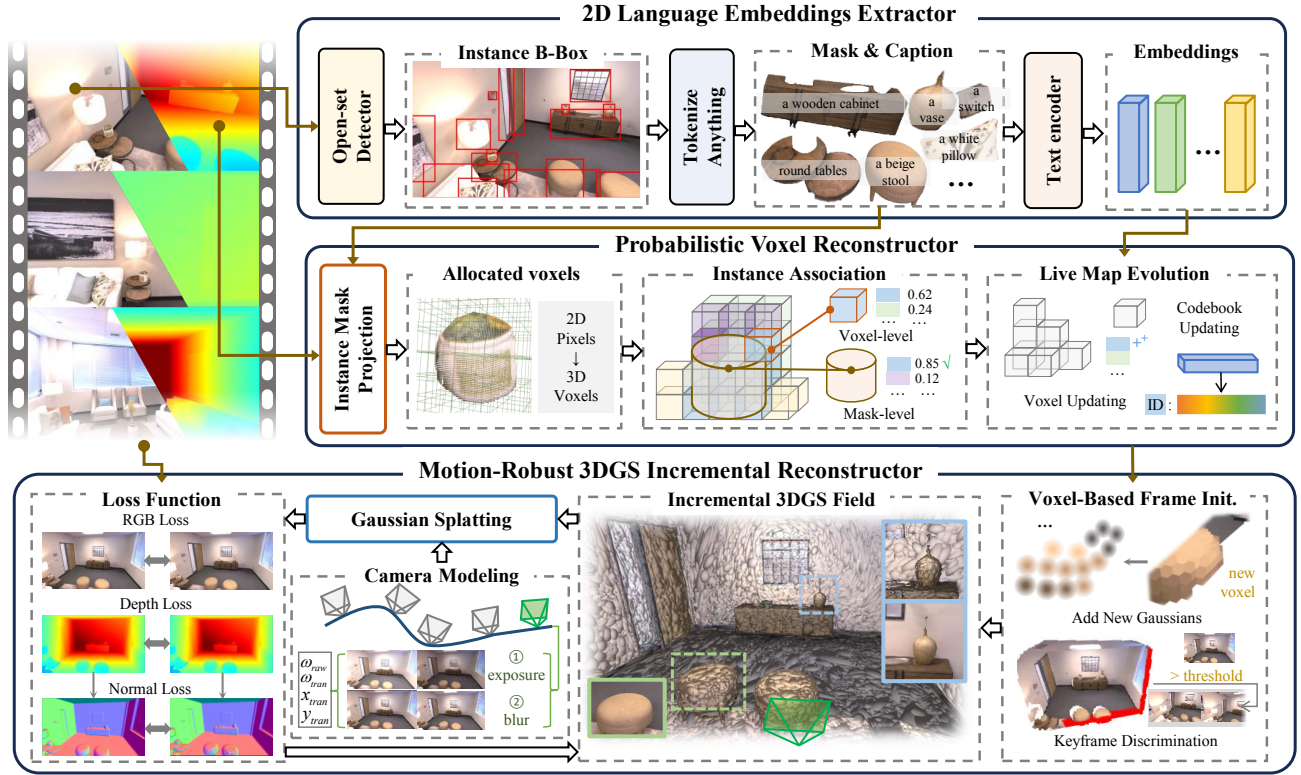


Fig. 3: The system framework of OmniMap consists of three main modules. The **2D Language Embeddings Extractor** sequentially combines multiple open-set models for detection, segmentation, captioning, and embedding extraction of objects. The **Probabilistic Voxel Reconstructor** incrementally integrates per-frame instance masks and embeddings into 3D space while maintaining voxel-aligned probabilistic instance tuples and a global embedding codebook. The **Motion-Robust 3DGS Incremental Reconstructor** initializes new Gaussians from new assigned voxels and renders 2D images under RGB, depth, and normal modalities with parameterized camera models for supervision.

III. OVERVIEW

OmniMap presents a general framework for robotic online mapping, enabling the simultaneous realization of optically, geometrically, and semantically accurate representations. OmniMap processes RGB-D video streams in real-time, including RGB image frames $\mathcal{C} = \{C_1, C_2, \dots, C_t\}$, depth image frames $\mathcal{D} = \{D_1, D_2, \dots, D_t\}$, and camera poses $\mathcal{P} = \{P_1, P_2, \dots, P_t\}$ (where t is the timestamp index).

As shown in Fig. 1, it constructs a complete 3DGS field \mathcal{G}_t consisting of Gaussian primitives $\mathcal{G}_t = \{g_t^i\}$ (where i is the Gaussian index) and open-vocabulary voxel map \mathcal{M}_t consisting of probabilistic instance-level voxels $\mathcal{V}_t = \{v_t^j\}$ (where j is the voxel index) and an embedding codebook $\mathcal{B}_t = \{f_t^\gamma \mid \gamma \in \Gamma\}$ (where γ is the instance index) that captures instance-level semantic understanding f_t^γ , where Γ is the set of all current instances. Fig. 3 illustrates the system framework, which consists of three core modules:

i) 2D Language Embeddings Extractor: This module extracts image embeddings for instance-level scene understanding in the 2D domain. OmniMap employs an efficient, robust pipeline that enhances semantic reasoning through the textual encoding of object captions.

ii) Probabilistic Voxel Reconstructor: This module performs incremental open-set instance fusion in the 3D domain. To address front-end segmentation noise, OmniMap employs a voxel-based probabilistic modeling framework.

iii) Motion-Robust 3DGS Incremental Reconstructor:

This module performs incremental 3DGS scene construction. By deriving new Gaussians and keyframes from voxels, OmniMap implements a lightweight and regularized Gaussian field. Meanwhile, camera parametric modeling and normal supervision improve the optical and geometric quality.

IV. 2D LANGUAGE EMBEDDINGS EXTRACTOR

Objects in open environments exhibit diverse types and structural distributions, posing challenges for classical deep learning-based semantic segmentation methods to maintain robust generalization across varied scenes. Recent advances in VLMs and LLMs offer new paradigms for open understanding. Unlike pixel-aligned feature extraction approaches [9], [20], [54], OmniMap adopts an instance-centric paradigm, treating instances as fundamental decomposable primitives for physical world representation, and focuses on capturing instance-level scene information.

A. Open-set Instance Detection and Segmentation

At time t , we first apply the real-time open-vocabulary detection model YOLO-World [58], denoted as $\text{Det}(\cdot)$, to detect instances in the image C_t . Detected targets with confidence scores exceeding a predefined threshold are localized via bounding boxes. Although YOLO-World provides coarse category predictions, it lacks fine-grained attributes (e.g., color, shape) and precise segmentation masks.

To address this, we seamlessly integrate the TAP model [59], denoted as $\text{SegCap}(\cdot)$, as a backend to YOLO-World. Prompted by YOLO-World’s bounding boxes, TAP concurrently produces pixel-level segmentation masks and corresponding descriptive captions. The output masks are denoted as $\{m_t^k\}$ (where k is the mask index) and the captions as $\{c_t^k\}$, encapsulating essential visual characteristics. To handle overlapping masks in the raw outputs, we apply a post-processing step that prioritizes smaller masks over larger ones and performs edge erosion. This process can be expressed as:

$$\{m_t^k, c_t^k\} = \text{SegCap}(\text{Det}(\mathbf{C}_t)) \quad (1)$$

B. LLM-Enhanced Embedding Encoding

The textual captions $\{c_t^k\}$, while human-interpretable, present challenges for robotic systems. Trained on large-scale corpora, LLMs demonstrate human-like reasoning capabilities. Inspired by this, OmniMap leverages LLMs $\text{Enc}(\cdot)$ to encode captions for enhanced semantic representations. Specifically, we utilize the SBERT model [60], which satisfies the critical requirement of mapping variable-length texts to fixed-dimensional embeddings. The resulting caption embeddings are denoted as $\{f_t^k\}$:

$$\{f_t^k\} = \text{Enc}(c_t^k) \quad (2)$$

After processing with above complete pipeline, we derive for the 2D image-size masks $\{m_t^k\}$ and caption embeddings $\{f_t^k\}$ from current RGB image \mathbf{C}_t , which will be grounded into 3D space in the next module.

V. PROBABILISTIC VOXEL RECONSTRUCTOR

After segmenting and comprehending the current frame \mathbf{C}_t , we perform incremental instance-level reconstruction to incorporate the results $\{m_t^k, f_t^k\}$ into the voxel map \mathcal{M}_{t-1} . Throughout this process, we adopt a probabilistic modeling framework to enhance the robustness of mapping, as shown in Fig. 4. The map is represented by probabilistic voxels \mathcal{V} , with each voxel v^j storing a probabilistic instance ID tuple θ^j . In practice, we apply the TSDF-Fusion algorithm [61] integrated with the Open3D framework. To optimize memory usage, the voxel map is composed of active hashing blocks, distributed only in the vicinity of obstacles, with each block containing 8^3 voxels. Open-vocabulary understanding is maintained via a separate embedding codebook \mathcal{B} that links each instance ID γ to its fused caption embedding f^γ .

A. Problem Definition and Decomposition

The incremental instance-level voxel mapping problem is defined as follows:

Given the current frame’s sensor observations and 2D instance cognition results $\mathcal{Q}_t = \{\{m_t^k, f_t^k\}, \mathbf{D}_t, \mathbf{P}_t\}$, determine \mathcal{I}_t the updated map $\mathcal{M}_t = \{\mathcal{V}_t, \mathcal{B}_t\}$ from the map at the last moment \mathcal{M}_{t-1} :

$$P(\mathcal{I}_t, \mathcal{M}_t | \mathcal{M}_{t-1}, \mathcal{Q}_t) \quad (3)$$

where $\mathcal{I}_t = \{\mathcal{I}_t^k\}$ represents the instance IDs assigned to all masks $\{m_t^k\}$, indicating their correspondence to the existing map instances Γ_{t-1} .



Fig. 4: The comparison between rendered and detected masks highlights its effectiveness in associating instances across frames (yellow lines) and mitigating missing, under-, or over-segmentation (red boxes).

Applying the chain rule, we derive the problem as:

$$\begin{aligned} P(\mathcal{I}_t, \mathcal{M}_t | \mathcal{M}_{t-1}, \mathcal{Q}_t) \\ = \underbrace{P(\mathcal{I}_t | \mathcal{M}_{t-1}, \mathcal{Q}_t)}_{\text{instance association}} \cdot \underbrace{P(\mathcal{M}_t | \mathcal{M}_{t-1}, \mathcal{Q}_t, \mathcal{I}_t)}_{\text{live map evolution}} \quad (4) \end{aligned}$$

This involves two subtasks: the instance association task $P(\mathcal{I}_t | \mathcal{M}_{t-1}, \mathcal{Q}_t)$ and the live map evolution task $P(\mathcal{M}_t | \mathcal{M}_{t-1}, \mathcal{Q}_t, \mathcal{I}_t)$.

B. Instance Association

Unlike traditional closed-set semantic frameworks, where semantics are inherently associated across frames, instance segmentation in open environments lacks mask inherent associations among frames, and the number of instances is unlimited. Furthermore, as shown in Fig. 4, instance segmentation quality fluctuates due to challenging conditions including uneven lighting, oblique viewpoints, and object clutter. Thus, robustly associating new-frame masks $\{m_t^k\}$ with existing instances Γ_{t-1} or initializing new ones becomes crucial.

Instance association involves mapping each segmented mask m_t^k to the instance ID set Γ_{t-1} from the current map \mathcal{M}_{t-1} :

$$P(\mathcal{I}_t | \mathcal{M}_{t-1}, \mathcal{Q}_t) = P(\mathcal{I}_t | \mathcal{V}_{t-1}, \mathcal{B}_{t-1}, \{m_t^k, f_t^k\}, \mathbf{D}_t, \mathbf{P}_t) \quad (5)$$

Given that mask overlaps are resolved in Subsection IV-A, each mask’s instance association can be treated independently. Under the conditional independence assumption, this reduces to solving separate association tasks for individual masks m_t^k :

$$\prod_i P(\mathcal{I}_t^k | \mathcal{V}_{t-1}, \mathcal{B}_{t-1}, m_t^k, f_t^k, \mathbf{D}_t, \mathbf{P}_t) \quad (6)$$

where \mathcal{I}_t^k is the map instance ID associated with mask m_t^k .

We first project the mask m_t^k into the voxel map \mathcal{V} according to the depth image \mathbf{D}_t to get the corresponding associated voxel region $V_{m_t^k}$:

$$V_{m_t^k} = \text{Vox}(\{\mathbf{D}_{t[u,v]} \mathbf{P}_t \mathbf{K}^{-1} \cdot [u, v] \mid [u, v] \in m_t^k\}) \quad (7)$$

where $[u, v]$ are the pixel coordinates within the mask m_t^k , \mathbf{K} is the camera internal parameter matrix, and $\text{Vox}(\cdot)$ is the 3D point-to-voxel transformation.

Algorithm 1: Instance Association Algorithm

Input: Segmented masks $\{m_t^k, f_t^k\}$;
Depth image \mathbf{D}_t , camera pose \mathbf{P}_t ;
Current voxel map \mathcal{V}_{t-1} and codebook \mathcal{B}_{t-1}

Output: Instance IDs $\{\mathcal{I}_t^k\}$ for each mask $\{m_t^k\}$

```

1 foreach mask  $m_t^k$  do
2   Project pixels  $[u, v] \in m_t^k$  to 3D using  $\mathbf{D}_t$  and  $\mathbf{P}_t$ 
3   Obtain corresponding voxel region  $V_{m_t^k}$  via Eq. (7)
4   if  $V_{m_t^k}$  has no prior instance then
5     Assign new instance ID  $\gamma_{\text{new}}$ , add to  $\Gamma_{t-1}$ 
6     Initialize  $\mathcal{B}_{t-1}[\gamma_{\text{new}}] \leftarrow f_t^k$ 
7   else
8     foreach existing instance  $\gamma \in \Gamma_{t-1}$  do
9       Compute geometric similarity  $S_{\mathcal{I}_t^k=\gamma}^{\text{geo}}$  via
10      Eq. (9)
11      Compute embedding similarity  $S_{\mathcal{I}_t^k=\gamma}^{\text{emb}}$  via
12      Eq. (10)
13      Fuse to get final score  $A_{\mathcal{I}_t^k=\gamma}$ 
14      if  $\max_{\gamma} A_{\mathcal{I}_t^k=\gamma} > \xi$  then
15        Assign  $\mathcal{I}_t^k = \arg \max_{\gamma} A_{\mathcal{I}_t^k=\gamma}$ 
16      else
17        Assign new instance ID  $\gamma_{\text{new}}$ , add to  $\Gamma_{t-1}$ 
18        Initialize  $\mathcal{B}_{t-1}[\gamma_{\text{new}}] \leftarrow f_t^k$ 
19   Update instance set:  $\Gamma_t \leftarrow \Gamma_{t-1}$ ;
20 return  $\{\mathcal{I}_t^k\}$ 

```

Consequently, our approach centers on computing the optimal instance assignment for the voxel set $V_{m_t^k}$. In cases where $V_{m_t^k}$ lacks prior instance information (indicating an initial observation), we perform two simultaneous operations: expanding the instance set Γ_{t-1} by creating a new instance γ_{new} , and initializing its associated embedding in the codebook \mathcal{B}_{t-1} using the extracted mask feature f_t^k . This approach efficiently handles previously unobserved instances while maintaining the integrity of the incremental mapping framework.

When $V_{m_t^k}$ contains existing instance information, the probability that the mask m_t^k is associated with these instances must be determined. Using a Bayesian formulation, we convert this task into a Maximum Likelihood Estimation (MLE) problem:

$$P(\mathcal{I}_t^k | V_{m_t^k}) \propto P(\mathcal{I}_t^k)P(V_{m_t^k} | \mathcal{I}_t^k) \quad (8)$$

where $P(\mathcal{I}_t^k)$ denotes the prior probability, assumed to be uniform across all instances Γ_{t-1} . Therefore, the goal is to solve for \mathcal{I}_t^k such that the likelihood of all voxels $v_{m_t^k}^j \in V_{m_t^k}$ having the current instance $\theta_{m_t^k}^j$ is maximized.

Since the voxels are relatively independent, a rigorous approach would involve multiplying the likelihood probabilities of each voxel. However, this introduces high computational complexity and is prone to arithmetic underflow. To mitigate this, we simplify the MLE process by accumulating evidence. By applying the law of large numbers, if the number of voxels $V_{m_t^k}$ is sufficiently large and the observed probability distri-

bution for each voxel is accurate, averaging these likelihood probabilities provides a reliable estimate of the geometric similarity $S_{\mathcal{I}_t^k=\gamma}^{\text{geo}}$ for mask m_t^k and instance γ :

$$S_{\mathcal{I}_t^k=\gamma}^{\text{geo}} = \mathbb{E}_j(\{\theta_{m_t^k}^j[\gamma]\}) \quad (9)$$

The geometric similarity metric $S_{\mathcal{I}_t^k=\gamma}^{\text{geo}}$ operates on the principle that spatially co-located masks are more likely to correspond to the same object. However, this assumption may break down for adjacent objects with tight spatial couplings, such as books on a table, especially in voxelized configurations. To address this limitation, we augment the geometric measure with embedding cosine similarity $S_{\mathcal{I}_t^k=\gamma}^{\text{emb}}$, enabling robust instance association through semantic-level feature matching:

$$S_{\mathcal{I}_t^k=\gamma}^{\text{emb}} = \text{Cos_Sim}(f_{t-1}^\gamma, f_t^k) \quad (10)$$

where f_{t-1}^γ is the embedding for instance γ in codebook \mathcal{B}_{t-1} .

The final association probability $A_{\mathcal{I}_t^k=\gamma}$ is obtained by the weighted fusion of the two similarities $S_{\mathcal{I}_t^k=\gamma}^{\text{geo}}$ and $S_{\mathcal{I}_t^k=\gamma}^{\text{emb}}$. Instance association occurs if the maximum probability $\text{Max}_{\gamma} (A_{\mathcal{I}_t^k=\gamma})$ exceeds the instance fusion threshold ξ ; otherwise, a new instance is initialized to Γ_{t-1} . The complete instance association algorithm is detailed in Alg. 1.

C. Live Map Evolution

The live map evolution jointly updates both the voxel representation \mathcal{V}_{t-1} and the embedding codebook \mathcal{B}_{t-1} . The voxel update process employs probabilistic local fusion to maintain robustness against segmentation noise, while the codebook progressively integrates multi-view instance perceptions through feature aggregation.

Voxel Updating: For voxel updating, we propose the Instance Counting Sensor Model (ICSM), inspired by the semantic counting sensor model in [62]. As previously mentioned, each voxel v^j in \mathcal{V} stores a probabilistic instance ID tuple $\theta^j = \{\theta^{j,\gamma} | \gamma \in \Gamma\}$, where $\theta^{j,\gamma} > 0$ and $\sum_{\gamma \in \Gamma} \theta^{j,\gamma} = 1$. During the instance association phase, we process observation data $\{\mathcal{V}_{m_t^k}, \mathcal{I}_t^k\}$, simplified as $\{(v_t^j, y_t^j)\}$, where v_t^j denotes the observed voxels in the current frame, and y_t^j is a one-hot-encoded measurement tuple representing the observed instance ID.

Voxel updating is essentially a Maximum A Posteriori estimation (MAP) task:

$$p(\theta_t^j | y_t^j) \propto p(y_t^j | \theta_t^j)p(\theta_t^j) \quad (11)$$

where the prior probability $p(\theta_t^j)$ is modeled as equivalent to the previous timestep's posterior probability $p(\theta_{t-1}^j)$. The likelihood $p(y_t^j | \theta_t^j)$ follows a Categorical distribution, representing the probability distribution over possible labels $y_t^{j,\gamma}$ for voxel v^j given the current observation:

$$p(y_t^j | \theta_t^j) = \prod_{\gamma \in \Gamma} (\theta_t^{j,\gamma})^{y_t^{j,\gamma}} \quad (12)$$

Algorithm 2: Live Map Evolution

Input: Observation set $\{(v_t^j, y_t^j)\}$;
Mask set $\{(m_t^k, \mathcal{I}_t^k, f_t^k)\}$;
Previous voxel map \mathcal{V}_{t-1} , Codebook \mathcal{B}_{t-1}

Output: Updated voxel map \mathcal{V}_t , Codebook \mathcal{B}_t

- 1 **foreach** voxel observation (v_t^j, y_t^j) **do**
- 2 $\alpha_t^j \leftarrow \alpha_{t-1}^j + y_t^j$
- 3 $\theta_{t,\gamma}^j \leftarrow \alpha_{t,\gamma}^j / \sum_{\gamma'} \alpha_{t,\gamma'}^j$
- 4 **foreach** mask m_t^k with instance ID \mathcal{I}_t^k and feature f_t^k **do**
- 5 Compute credibility w_t^k via Eq. (17) and Eq. (18)
- 6 $W_t^{\mathcal{I}_t^k} \leftarrow W_{t-1}^{\mathcal{I}_t^k} + w_t^k$
- 7 $f_t^{\mathcal{I}_t^k} \leftarrow (W_{t-1}^{\mathcal{I}_t^k} f_{t-1}^{\mathcal{I}_t^k} + w_t^k f_t^k) / W_t^{\mathcal{I}_t^k}$
- 8 Update voxel map: $\mathcal{V}_t \leftarrow \mathcal{V}_{t-1}$
- 9 Update codebook: $\mathcal{B}_t \leftarrow \mathcal{B}_{t-1}$
- 10 **return** $\mathcal{V}_t, \mathcal{B}_t$

When applying the Dirichlet distribution, the conjugate prior for the Categorical distribution, to the prior probability, the posterior probability remains of the same distribution type:

$$p(\theta_{t-1}^j) \propto \prod_{\gamma \in \Gamma} (\theta_{t-1}^j)^\gamma \alpha_{t-1}^{j,\gamma-1} \quad (13)$$

$$p(\theta_t^j | y_t^j) \propto \prod_{\gamma \in \Gamma} (\theta_t^j)^\gamma \alpha_t^{j,\gamma-1} \quad (14)$$

where α_{t-1}^j and α_t^j are the concentration parameters of the prior and posterior, respectively.

Substituting Eq. (12), Eq. (13) and Eq. (14) into (11), we can deduce that:

$$\alpha_t^{j,\gamma} = \alpha_{t-1}^{j,\gamma} + y_t^{j,\gamma} \quad (15)$$

Since $\alpha_t^{j,\gamma}$ counts the number of times voxel v^j is associated with instance label γ , we refer to this model as the instance counting sensor model. Given α_t^j , the probabilistic instance tuple of the voxel v^j is the closed-form expected value of the posterior Dirichlet [14]:

$$\theta_t^{j,\gamma} = \frac{\alpha_t^{j,\gamma}}{\sum_{\gamma' \in \Gamma} \alpha_t^{j,\gamma'}} \quad (16)$$

The Dirichlet concentration parameters α_t^j act as pseudo-counts that accumulate evidence over time. This formulation provides three key properties: (i) the update rule Eq. (15) becomes a simple counting operation, (ii) the expected value Eq. (16) yields interpretable probabilities, and (iii) the variance naturally decreases with accumulating evidence.

Codebook Updating: For the codebook updating, we use a weighted fusion strategy. For each mask m_t^k , the associated instance ID \mathcal{I}_t^k is obtained in the instance association step. The credibility w_t^k of its current frame observation features f_t^k is evaluated by combining the association probability $A_{\mathcal{I}_t^k}$ and the visibility ratio R_t^k :

$$w_t^k = A_{\mathcal{I}_t^k} \cdot R_t^k \quad (17)$$

$$R_t^k = \frac{|V_{m_t^k}|}{\left| \left\{ \arg \max_{\gamma} (\theta_{t-1}^j[\gamma]) = \mathcal{I}_t^k \right\}_j \right|} \quad (18)$$

The visibility ratio R_k represents the proportion of the instance's size observed by the current mask m_t^k relative to the total size of the instance. This helps prevent mask features with poor viewing (e.g., only a corner of a couch is visible) from contaminating the global codebook. Based on this, the updating of the codebook can be derived as:

$$f_t^{\mathcal{I}_t^k} = (W_{t-1}^{\mathcal{I}_t^k} f_{t-1}^{\mathcal{I}_t^k} + w_t^k f_t^k) / W_t^{\mathcal{I}_t^k} \quad (19)$$

$$W_t^{\mathcal{I}_t^k} = W_{t-1}^{\mathcal{I}_t^k} + w_t^k \quad (20)$$

where $f_t^{\mathcal{I}_t^k}$ and $W_t^{\mathcal{I}_t^k}$ are the updated features and weights of instance \mathcal{I}_t^k in embedding codebook \mathcal{B}_t , respectively. The complete instance association algorithm is detailed in Alg. 2.

VI. MOTION-ROBUST 3DGS INCREMENTAL RECONSTRUCTOR

The preceding sections have robustly achieved incremental scene understanding through voxel-based representations. While voxels enable locally fused updates, their inherent resolution limitations fundamentally restrict the ability to reconstruct high-fidelity geometric and photometric details of the scene. To address these limitations, we now introduce an incremental representation based on 3DGS [6], which reconstructs scene observations using more flexible and differentiable Gaussian primitives. This approach overcomes the rigid structural constraints of voxel grids while maintaining the incremental update capabilities essential for scene reconstruction.

A. Preliminary for 3DGS

3DGS [6] is composed of 3D explicit Gaussian ellipsoids $\{g^i\}$ parameterized by

- 1) *3D center*: $\mu^i = (x^i, y^i, z^i) \in \mathbb{R}^3$
- 2) *3D rotation*: $q^i = (qw^i, qx^i, qy^i, qz^i) \in \mathbb{R}^4$
- 3) *RGB color*: $c^i = (r^i, g^i, b^i) \in \mathbb{R}^3$
- 4) *3D size (scaling factor)*: $s^i = (sx^i, sy^i, sz^i) \in \mathbb{R}^3$
- 5) *Opacity logit*: $o^i \in [0, 1]$

Each Gaussian occupies a spatial region, and RGB values c^i can alternatively be represented as view-dependent spherical harmonics coefficients. For an arbitrary query point $x \in \mathbb{R}^3$, the i -th Gaussian g^i 's contribution is weighted by its opacity o^i , computed via the standard Gaussian function:

$$\eta^i(x) = o^i \exp\left(-\frac{1}{2}(x - \mu^i)\Sigma^i(x - \mu^i)\right) \quad (21)$$

where $\Sigma^i \in \mathbb{R}^{3 \times 3}$ is the covariance matrix calculated by combining the scaling component $S^i = \text{darg}([sx^i \ sy^i \ sz^i])$ and the rotation component $R^i = \text{q2R}([qw^i \ qx^i \ qy^i \ qz^i])$.

Due to their formulation, Gaussians have theoretically infinite support, which leads to gradients having a global influence during optimization. To fit observed scenes, 3DGS differentially renders Gaussians $\{g^i\}$ into images. During rendering,

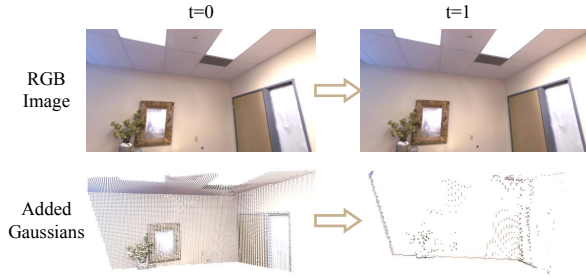


Fig. 5: OmniMap initializes the Gaussian by distinguishing the newly added voxels, providing an efficient yet effective coverage of newly observed regions. After initialization, only minimal Gaussian additions are required for subsequent updates.

Gaussian primitives are depth-sorted and projected via a 2D version of Eq. (21). The 2D density of g^i is denoted as α^i . The color $\hat{\mathbf{C}}(p)$ and depth $\hat{\mathbf{D}}(p)$ of each pixel p are then computed using alpha-blending:

$$\hat{\mathbf{C}}(p) = \sum_i c^i \alpha^i \prod_{j=1}^{i-1} (1 - \alpha^j) \quad (22)$$

$$\hat{\mathbf{D}}(p) = \sum_i h^i \alpha^i \prod_{j=1}^{i-1} (1 - \alpha^j) \quad (23)$$

where h^i represent the depth of the Gaussian primitive g^i in given camera pose.

B. Voxel-Based Frame Initialization

Add New Gaussians: Unlike the original 3DGS paradigm, online incremental 3DGS requires the continuous addition of new Gaussians to adapt to newly observed regions, which are unpredictable a priori. Some existing methods [53] adopt a brute-force incremental approach, back-projecting each keyframe to initialize Gaussian centers from point clouds. However, this leads to uneven scene distribution, where frequently observed regions accumulate excessive Gaussian density compared to rarely viewed areas. To mitigate this, alternative methods [17] use rendered alpha-blending values to identify newly observed pixels for back-projection. Yet, this approach remains vulnerable to artifacts from large-scale Gaussians in novel views and lacks 3D physical consistency.

Thanks to TSDF-Fusion’s standardized voxel representation, OmniMap achieves regular and efficient Gaussian initialization. Specifically, OmniMap identifies candidate voxels for Gaussian addition through a simple filter operation on the counting values from Eq. (15):

$$\mathcal{V}_t^{\text{new}} = \left\{ v^j \mid \alpha_{t-1}^j = 0 \text{ and } \alpha_t^j > 0 \right\} \quad (24)$$

$\mathcal{V}_t^{\text{new}}$ is the new voxel centers, from where new Gaussians $\mathcal{G}_t^{\text{new}}$ are initialized. The colors of $\mathcal{G}_t^{\text{new}}$ are set as the voxel colors from TSDF-Fusion. Their spatial extent s is scaled to 0.2 times the voxel size, with an initial opacity o of 0.5. Fig. 5 demonstrates OmniMap’s Gaussian addition process in a representative case.

Keyframe Discrimination: Similar to other 3DGS-based SLAM systems, OmniMap maintains a keyframe buffer to prevent catastrophic forgetting during incremental optimization.

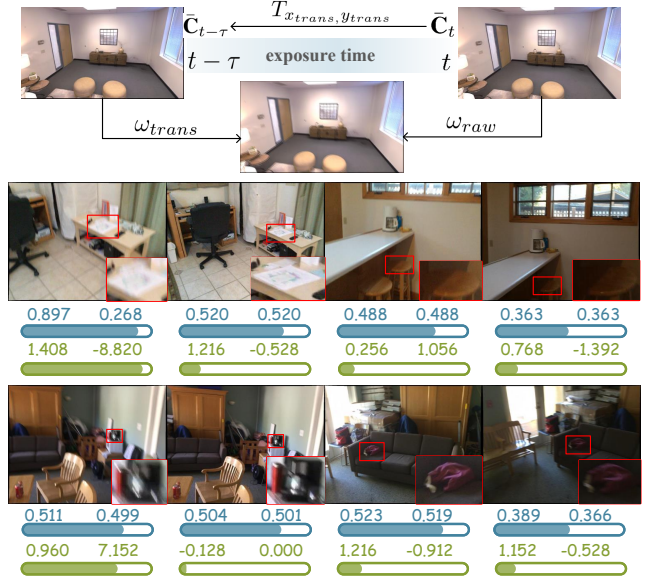


Fig. 6: Schematic of camera modeling. The numbers under images correspond to the optimized parameter set $[\omega_{raw}, \omega_{trans}, x_{trans}, y_{trans}]$. Quantitative analysis reveals: (1) high-brightness images show elevated $\omega_{raw} + \omega_{trans}$ (blue bar) relative to low-brightness conditions, and (2) blurred images exhibit consistently higher $|x_{trans}| + |y_{trans}|$ (green bar) compared to sharp counterparts. These correlations validate OmniMap’s effectiveness in camera modeling.

To ensure effective scene coverage, keyframe selection prioritizes maximally divergent viewpoints, thereby enhancing the quality of multi-view reconstruction. Our approach quantifies view dissimilarity by computing the ratio of unregistered to observed regions at the voxel level. To do this, we introduce a boolean variable $r(v_t^j)$ for each voxel v^j , initialized as unregistered and updated to registered upon keyframe inclusion. When the unregistered region ratio τ_t exceeds a predefined threshold, the current frame is added to the keyframe buffer:

$$\tau_t = \frac{1}{n} \sum_{j=1}^n \mathbb{I}(r(v_t^j)) \quad (25)$$

where n denotes the total number of observable voxels at time t with pose \mathbf{P}_t , and \mathbb{I} is the indicator function. In OmniMap, keyframes are selected not only based on the unregistered ratio but also when more than N_{key} consecutive frames are observed without inserting a new keyframe. Such temporal density control avoids under-sampling of the scene, which is critical for dense reconstruction accuracy.

C. Camera Modeling

In image-based multi-view dense mapping, the camera model plays a critical role in determining reconstruction quality. The fundamental camera parameters consist of intrinsic and extrinsic parameters. During incremental 3DGS mapping, intrinsic parameters \mathbf{K} remain fixed (as they are camera-specific), while extrinsic parameters are derived from the camera pose \mathcal{P} , typically estimated by visual SLAM systems. However, two pervasive challenges extend beyond these basic parameters: motion blur and exposure inconsistency. Motion

blur arises from relative camera-scene motion during exposure, resulting in smeared trajectories in captured images. Exposure inconsistency occurs when varying lighting conditions or view-dependent reflectivity cause divergent brightness levels across different viewpoints, more precisely, commonly perceived as variations in light intensity.

The physical process of camera motion blur can be mathematically modeled as a temporal integration of latent sharp images along the camera’s trajectory during exposure:

$$\mathbf{C}_t = \phi \int_{t-\tau}^t \tilde{\mathbf{C}}_t dt \quad (26)$$

where \mathbf{C}_t denotes the real captured motion-blurred image at time t , ϕ serves as a normalization factor, $\tilde{\mathbf{C}}$ is the virtual latent sharp image captured at timestamp $[t - \tau, t]$ within the exposure time.

To simplify the solution, OmniMap discretely approximates the model in Eq. (26) as a weighted average of the exposure start and end moments:

$$\mathbf{C}_t \approx \omega_{trans} \tilde{\mathbf{C}}_{t-\tau} + \omega_{raw} \tilde{\mathbf{C}}_t \quad (27)$$

where ω_{trans} and ω_{raw} are two control parameters in the range $[0, 1]$. With this setting, the camera model in Eq. (27) can effectively simulate exposure intensity: when $\omega_{trans} + \omega_{raw} > 1$, the image appears brighter; and when $\omega_{trans} + \omega_{raw} < 1$, the image appears darker.

$\tilde{\mathbf{C}}_{t-\tau}$ represents a simulated sharp image at a discrete timestep within the current frame’s exposure period. For small time intervals, minor camera translation can be approximated as a global image shift, neglecting depth variations and rotational effects. We model $\tilde{\mathbf{C}}_{t-\tau}$ using the image transformation parameters x_{trans} and y_{trans} :

$$\tilde{\mathbf{C}}_{t-\tau} \approx T_{x_{trans}, y_{trans}}(\tilde{\mathbf{C}}_t) \quad (28)$$

where $T_{x_{trans}, y_{trans}}(\cdot)$ denotes a translation transformation $[x_{trans}, y_{trans}]$ for the image. Thus the rendered image is modeled as four parameters:

$$\mathbf{C}_t \approx \omega_{trans} T_{x_{trans}, y_{trans}}(\tilde{\mathbf{C}}_t) + \omega_{raw} \tilde{\mathbf{C}}_t \quad (29)$$

Similar to the Gaussian primitives, these four parameters $[\omega_{raw}, \omega_{trans}, x_{trans}, y_{trans}]$ are jointly optimized within the 3DGS pipeline, which are initialized to $[0.5, 0.5, 0, 0]$. The incorporation of camera modeling significantly enhances scene reconstruction quality while mitigating the impact of motion blur and exposure variations on Gaussian field optimization. As shown in Fig. 6, our camera modeling approach is presented alongside experimental results that verify the effectiveness of the proposed blur and exposure modeling strategy.

D. Loss Function

To ensure Gaussian primitives faithfully reconstruct the real-world scene and the camera model accurately represent the imaging process, OmniMap employs multiple loss functions to optimize these parameters.

RGB Loss: For keyframe with time t' , we employ Eq. (22) to render the raw color image $\hat{\mathbf{C}}_{t'}$. To ensure sharp reconstruction, we transform the rendered image $\hat{\mathbf{C}}_{t'}$ using

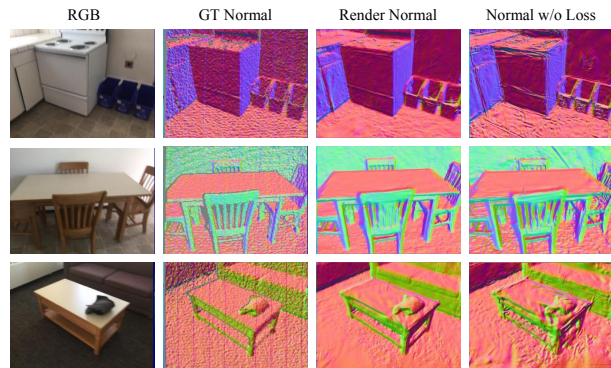


Fig. 7: OmniMap’s rendering produces smooth and geometrically consistent normal images despite being trained with noisy supervision, demonstrating significantly better reconstruction quality than approaches without \mathcal{L}_{normal} .

Eq. (29) with the corresponding camera model parameters for current keyframe, producing $\hat{\mathbf{C}}_{t'}^{cam}$, which is then supervised against the real observed image $\mathbf{C}_{t'}$. This supervision is implemented through a combination of L1 loss \mathcal{L}_{rgb} and SSIM loss \mathcal{L}_{ssim} :

$$\mathcal{L}_{rgb} = \left\| \hat{\mathbf{C}}_{t'}^{cam} - \mathbf{C}_{t'} \right\|_1 \quad (30)$$

$$\mathcal{L}_{ssim} = \text{SSIM}(\hat{\mathbf{C}}_{t'}^{cam}, \mathbf{C}_{t'}) \quad (31)$$

Depth Loss: Relying solely on RGB supervision may lead to overfitting, significantly degrading novel view synthesis quality. To address this, we incorporate depth supervision to guide Gaussian primitives in accurately conforming to object surfaces, thereby improving geometric fidelity. The depth supervision is implemented using an L1 loss \mathcal{L}_{depth} between the rendered depth $\hat{\mathbf{D}}_{t'}$ and ground truth depth $\mathbf{D}_{t'}$:

$$\mathcal{L}_{depth} = \left\| \hat{\mathbf{D}}_{t'} - \mathbf{D}_{t'} \right\|_1 \quad (32)$$

Normal Loss: Compared to depth, normals exhibit stronger local geometric sensitivity and multi-view consistency. Departing from monocular estimation approaches [53], we directly compute normal images from depth images. To enforce geometric accuracy, we employ a cosine similarity loss \mathcal{L}_{normal} between the rendered normal image $\hat{\mathbf{N}}_{t'}$ (derived from $\hat{\mathbf{D}}_{t'}$) and the ground truth normal image $\mathbf{N}_{t'}$ (derived from $\mathbf{D}_{t'}$):

$$\mathcal{L}_{normal} = 1 - \langle \hat{\mathbf{N}}_{t'}, \mathbf{N}_{t'} \rangle \quad (33)$$

where $\langle \cdot \rangle$ denotes the inner product operation. As illustrated in Fig. 7, the optimized rendered normal image $\hat{\mathbf{N}}_{t'}$ demonstrates superior geometric fidelity after optimization, despite noise-induced artifacts in the ground truth normal image $\mathbf{N}_{t'}$ derived from depth measurements. This improvement is particularly evident when compared to reconstruction results without normal supervision, where fine geometric details are significantly harder to recover.

By integrating the aforementioned losses through a weighted combination, OmniMap progressively optimizes both the Gaussian primitives and camera parameters. This optimization framework enables incremental reconstruction of scene appearance and geometry, facilitated by the dynamically expanding keyframe collection.

TABLE II: Rendering quality evaluations on Replica dataset.

Method	Metric	ro-0	ro-1	ro-2	of-0	of-1	of-2	of-3	of-4	Avg.
RTG-SLAM	PSNR \uparrow	26.14	30.64	31.56	37.32	37.43	30.65	30.54	33.37	32.21
	SSIM \uparrow	0.905	0.947	0.956	0.985	0.986	0.968	0.970	0.974	0.961
	LPIPS \downarrow	0.281	0.194	0.188	0.094	0.112	0.180	0.167	0.161	0.172
Gaussian-SLAM	PSNR \uparrow	32.34	33.08	35.08	41.16	42.73	34.28	35.76	36.55	36.37
	SSIM \uparrow	0.913	0.921	0.953	0.980	0.982	0.951	0.963	0.967	0.954
	LPIPS \downarrow	0.100	0.107	0.063	0.025	0.026	0.079	0.048	0.056	0.063
MonoGS	PSNR \uparrow	33.20	34.80	36.41	39.52	40.73	35.91	36.04	33.23	36.23
	SSIM \uparrow	0.939	0.942	0.964	0.969	0.976	0.966	0.968	0.938	0.958
	LPIPS \downarrow	0.073	0.060	0.041	0.034	0.031	0.035	0.028	0.077	0.047
SplaTAM	PSNR \uparrow	33.42	33.86	35.32	39.57	40.56	32.15	30.88	32.71	34.81
	SSIM \uparrow	0.982	0.974	0.988	0.989	0.987	0.971	0.961	0.958	0.976
	LPIPS \downarrow	0.059	0.067	0.045	0.040	0.052	0.074	0.086	0.105	0.066
GSFusion	PSNR \uparrow	29.797	31.81	34.83	38.04	38.05	31.33	32.58	36.28	34.70
	SSIM \uparrow	0.929	0.948	0.965	0.970	0.957	0.953	0.960	0.972	0.961
	LPIPS \downarrow	0.063	0.076	0.055	0.043	0.079	0.086	0.042	0.044	0.061
GS-ICP-SLAM	PSNR \uparrow	35.24	36.24	36.94	41.06	41.15	34.72	34.94	39.52	37.47
	SSIM \uparrow	0.962	0.961	0.971	0.985	0.982	0.969	0.972	0.979	0.973
	LPIPS \downarrow	0.035	0.049	0.041	0.020	0.028	0.055	0.042	0.028	0.037
CaRtGS	PSNR \uparrow	31.92	31.84	37.59	36.00	37.28	33.43	34.27	36.75	34.88
	SSIM \uparrow	0.922	0.924	0.970	0.954	0.956	0.943	0.946	0.961	0.947
	LPIPS \downarrow	0.058	0.077	0.029	0.060	0.038	0.070	0.059	0.043	0.054
HI-SLAM2	PSNR \uparrow	33.37	34.96	36.17	40.14	41.11	35.23	34.93	36.76	36.59
	SSIM \uparrow	0.945	0.957	0.966	0.979	0.981	0.962	0.962	0.965	0.965
	LPIPS \downarrow	0.054	0.052	0.043	0.030	0.025	0.053	0.046	0.053	0.044
Ours	PSNR \uparrow	37.93	38.27	38.68	42.01	42.06	38.23	38.29	40.38	39.48
	SSIM \uparrow	0.979	0.978	0.979	0.989	0.983	0.980	0.981	0.985	0.982
	LPIPS \downarrow	0.020	0.027	0.028	0.013	0.024	0.021	0.018	0.016	0.021

VII. BENCHMARK RESULTS

A. Experimental Setup

Implementation Details: Our implementation primarily utilizes the PyTorch framework and is tested on a single RTX 4090 GPU. In all experiments, we set the resolution of the voxels to 0.03m to balance precision and memory. For the initial frame, we execute 1000 warm-up iterations. For subsequent frames, we process each input frame once, along with 19 randomly selected keyframes from the buffer. No Gaussian densification or pruning is performed beyond the voxel-based frame initialization.

Baseline: We benchmark OmniMap against SOTA methods in both 3DGS incremental mapping and open-vocabulary understanding. For rendering and reconstruction evaluations, we compare it with 8 3DGS incremental mapping methods (RTG-SLAM [49], Gaussian-SLAM [17], MonoGS [19], SplaTAM [18], GSFusion [22], GS-ICP-SLAM [51], CaRtGS [52], and HI-SLAM2 [53]) providing all methods with ground-truth poses \mathcal{P} for fair comparison. For open-vocabulary semantic segmentation, we evaluate it against 3 3DGS-based approaches (OpenGaussian [57], LangSplat [10], and GraspSplats [63]) and 4 incremental methods (ConceptFusion [20], ConceptGraph [7], OpenFusion [28], and HOV-SG [8]). For fair comparison, we reproduced all baseline methods using their official implementations in our local environment.

Dataset and Metrics: We select eight scenes from the synthetic Replica [64] dataset and six scenes from the real-world ScanNet [65] dataset to represent a diverse set of environments. For rendering quality evaluation, we employ standard metrics: PSNR, SSIM, and LPIPS. Geometric re-

TABLE III: Rendering quality evaluations on ScanNet dataset.

Method	Metric	0000	0059	0106	0169	0181	0207	Avg.
RTG-SLAM	PSNR \uparrow	19.07	17.45	15.83	18.88	18.17	19.27	18.11
	SSIM \uparrow	0.820	0.747	0.712	0.792	0.776	0.790	0.773
	LPIPS \downarrow	0.469	0.500	0.548	0.471	0.551	0.488	0.504
Gaussian-SLAM	PSNR \uparrow	23.74	21.51	22.09	25.03	22.95	24.75	23.35
	SSIM \uparrow	0.754	0.739	0.785	0.789	0.811	0.785	0.777
	LPIPS \downarrow	0.475	0.430	0.418	0.401	0.519	0.450	0.449
MonoGS	PSNR \uparrow	19.60	18.13	17.73	20.42	17.93	19.48	18.88
	SSIM \uparrow	0.696	0.663	0.687	0.726	0.743	0.710	0.070
	LPIPS \downarrow	0.499	0.454	0.527	0.455	0.519	0.477	0.049
SplaTAM	PSNR \uparrow	19.19	19.09	17.99	22.21	16.72	19.31	19.09
	SSIM \uparrow	0.655	0.773	0.700	0.789	0.670	0.685	0.712
	LPIPS \downarrow	0.427	0.290	0.350	0.285	0.416	0.349	0.353
GSFusion	PSNR \uparrow	18.98	18.86	19.22	19.29	17.79	20.04	19.03
	SSIM \uparrow	0.733	0.764	0.811	0.708	0.745	0.769	0.755
	LPIPS \downarrow	0.419	0.355	0.353	0.389	0.491	0.437	0.407
GS-ICP-SLAM	PSNR \uparrow	21.16	19.53	18.19	20.57	21.02	21.57	20.34
	SSIM \uparrow	0.762	0.759	0.782	0.784	0.830	0.772	0.782
	LPIPS \downarrow	0.438	0.429	0.440	0.402	0.493	0.477	0.447
CaRtGS	PSNR \uparrow	19.57	20.00	19.24	20.31	17.53	18.04	19.12
	SSIM \uparrow	0.759	0.812	0.822	0.796	0.792	0.692	0.779
	LPIPS \downarrow	0.397	0.353	0.364	0.362	0.590	0.535	0.433
HI-SLAM2	PSNR \uparrow	24.10	21.33	22.40	26.53	22.23	23.65	23.37
	SSIM \uparrow	0.785	0.793	0.825	0.805	0.829	0.774	0.802
	LPIPS \downarrow	0.368	0.341	0.331	0.372	0.424	0.429	0.378
Ours	PSNR \uparrow	26.61	24.30	25.05	27.05	26.33	26.90	26.04
	SSIM \uparrow	0.813	0.831	0.860	0.836	0.873	0.812	0.838
	LPIPS \downarrow	0.326	0.284	0.284	0.278	0.322	0.353	0.308

construction is assessed using Accuracy (Acc.), completeness (Comp.), completion rate (Comp. Rat.), and F-score. For open-vocabulary semantic understanding, we measured: mean IoU (mIoU), frequency-weighted IoU (fIoU), mean accuracy (mAcc), frequency-weighted accuracy (fAcc).

B. Optics: Image Rendering

To evaluate the optical reconstruction capability of OmniMap, we rendered novel views using camera trajectories \mathcal{P} from the reconstructed 3DGS field \mathcal{G} representation and quantitatively compared them with GT RGB images \mathcal{C} . We present quantitative rendering results for the Replica and ScanNet datasets in Tab. II and Tab. III, respectively. OmniMap achieves the best performance across almost all synthetic and real scenes, owing to several key design features: voxel-based Gaussian initialization, which ensures ample optimization potential in any region; a small number of iterations over multiple keyframes, enabling collaborative supervision from various viewpoints; and modeling of camera parameters to address motion blur and exposure inconsistencies across frames. The performance gain is particularly significant on ScanNet’s challenging real-world scenes with frequent blurring and exposure issues. While alternative methods show competitive performance in specific scenarios, OmniMap consistently outperforms across metrics.

Qualitative results for representative scenes are shown in Fig. 8. In the noise-free Replica dataset (top four rows), all methods are nearly indistinguishable from the true images. However, when zooming in on details such as the stripes on the floor stand and the mesh chair backrest, OmniMap

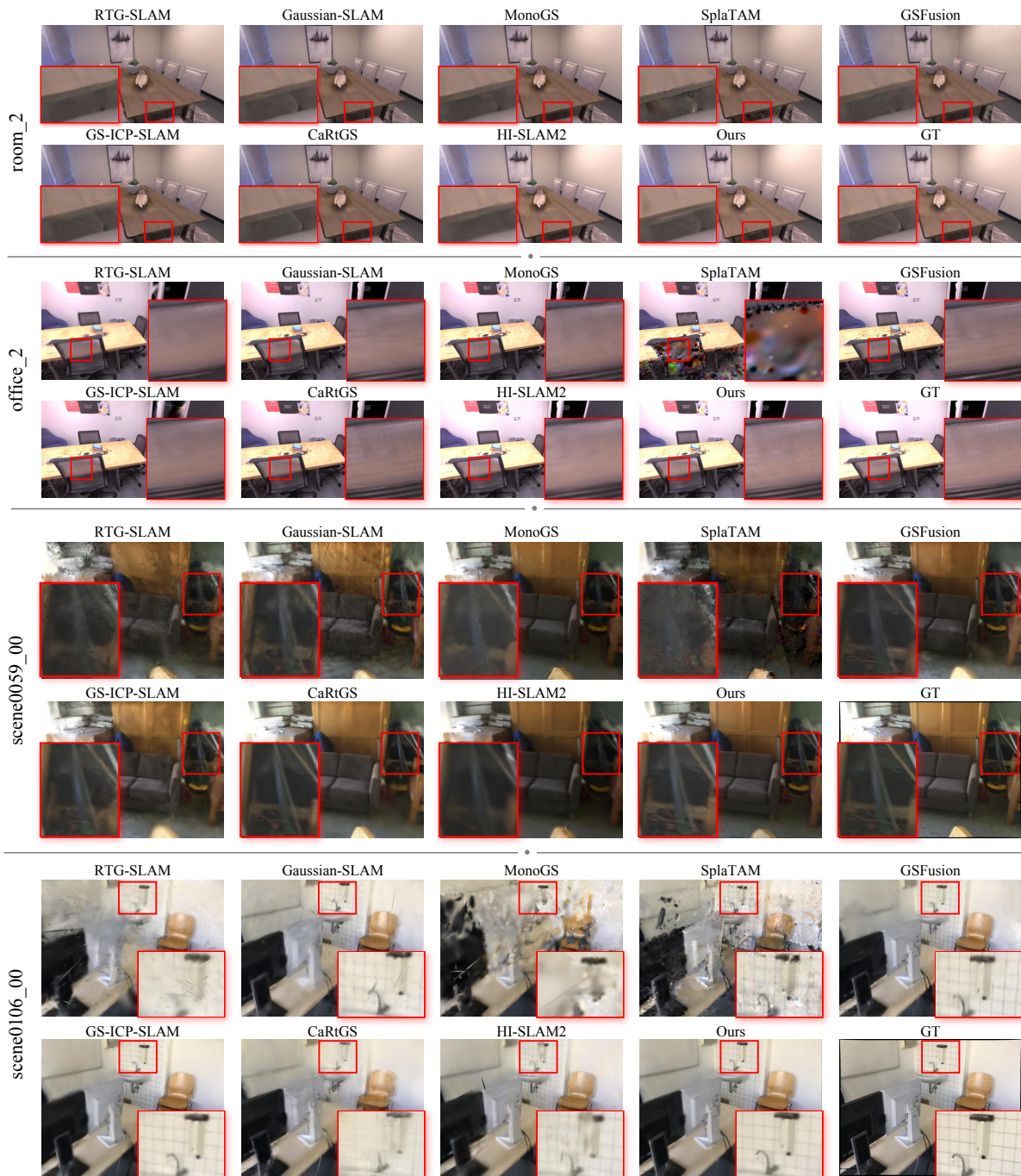


Fig. 8: The rendering comparisons across diverse scenes demonstrate OmniMap’s superior visual quality, achieving both higher overall fidelity and more accurate texture details than all baseline methods.

exhibits higher fidelity in color, indicating superior detailing capabilities in optical reconstruction. The accurate reproduction of fine textures further emphasizes OmniMap’s ability to capture subtle features, which may be missed by competing methods. The ScanNet comparisons (bottom rows) provide more compelling evidence, with OmniMap accurately reconstructing challenging elements (long sticks, wall meshes) that other methods either distort or fail to capture completely. Notably, OmniMap excels at reconstructing long, thin objects and intricate details that often suffer from noise or occlusion in real-world datasets. This highlights the method’s robust-

ness and flexibility in handling diverse scene geometries and textures. OmniMap even demonstrates superior reconstruction quality compared to GT in certain scenarios, as evidenced by scene0059_00 where our method recovers clearer rope textures that appear blurred in the original color observation.

C. Geometry: Mesh Reconstruction

To obtain the meshes, we rendered depth images from 3DGS and performed TSDF-Fusion at a consistent resolution of 0.01m for all methods. While HI-SLAM2 incorporates monocular depth estimation, as a monocular 3DGS method,

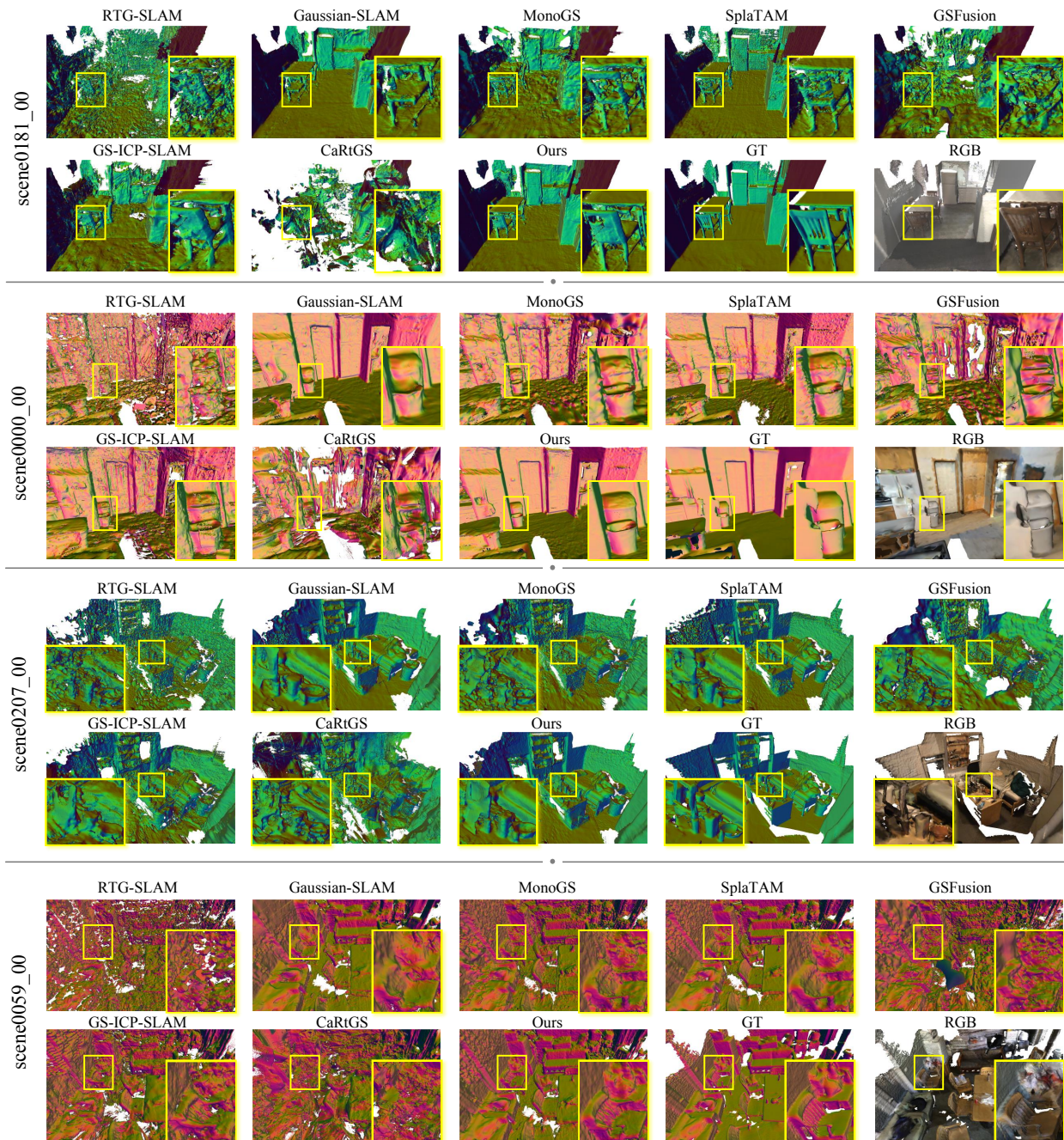


Fig. 9: Mesh reconstruction quality comparison. OmniMap generates notably smoother surfaces and more geometrically precise reconstructions, particularly excelling in both continuous surface representation and fine structural detail recovery.

it is excluded from geometric evaluation. Furthermore, we conduct validation exclusively on the ScanNet dataset, as all methods achieve near-optimal performance on Replica.

The quantitative results are summarized in Tab. IV. All methods demonstrate generally worse Acc. than Comp., attributable to ScanNet’s conservative ground truth mesh that excludes certain regions. Despite this, it maintains strong performance through its canonical Gaussian initialization and higher-order geometric signal supervision. In contrast, GSFusion and CaRtGS produce the poorest results, as they prioritize rendering quality over effective depth image supervision, leading to suboptimal mesh reconstruction.

Qualitative results are displayed in Fig. 9. All methods except OmniMap show misaligned artifacts and uneven surfaces, primarily caused by noisy depth images and inter-frame depth inconsistencies. OmniMap inherits the effect of normal consistency across viewpoints, producing smoother meshes. The close-up visualization further illustrates OmniMap’s advantage in reconstructing thin structures, such as chair legs, and continuous geometries, like garbage cans. Notably, OmniMap surpasses GT mesh quality in certain cases (e.g., the display screen in scene0207_00) by inferring depth values from multi-view RGB observations that appear missing in the original depth observations.

TABLE IV: Reconstruction quality evaluations on ScanNet dataset.

Method	Metric	0000	0059	0106	0169	0181	0207	Avg.
RTG-SLAM	Acc. ↓	03.98	13.67	22.84	05.23	08.83	08.51	10.51
	Comp. ↓	02.54	04.00	04.57	02.86	05.69	02.70	03.73
	Comp. Rat. ↑	94.26	82.70	76.22	90.19	61.94	90.06	82.56
	F-Score ↑	85.86	61.72	54.18	78.40	48.82	78.59	67.93
Gaussian-SLAM	Acc. ↓	04.21	13.54	25.98	07.86	05.12	09.90	11.10
	Comp. ↓	03.93	05.35	04.34	04.94	04.52	03.86	04.49
	Comp. Rat. ↑	75.70	57.83	68.07	52.57	61.62	79.72	65.92
	F-Score ↑	75.46	51.07	50.31	50.14	61.58	72.87	60.24
MonoGS	Acc. ↓	04.78	11.02	25.33	06.91	05.82	08.97	10.47
	Comp. ↓	03.91	03.66	04.45	03.05	04.20	02.76	03.67
	Comp. Rat. ↑	86.37	84.45	76.76	86.99	75.19	89.38	83.19
	F-Score ↑	82.50	70.09	53.53	77.80	71.66	77.62	72.20
SplaTAM	Acc. ↓	03.02	12.73	26.11	09.67	03.96	09.61	10.85
	Comp. ↓	02.55	03.15	03.74	03.33	03.04	02.62	03.07
	Comp. Rat. ↑	91.04	84.95	79.68	84.41	86.62	91.21	86.32
	F-Score ↑	89.77	71.28	58.29	73.17	84.66	81.37	76.42
GSFusion	Acc. ↓	07.45	12.35	26.97	12.24	10.50	11.46	13.50
	Comp. ↓	04.80	04.24	07.89	07.40	07.67	05.82	06.30
	Comp. Rat. ↑	68.39	72.91	47.62	43.11	44.32	60.94	56.22
	F-Score ↑	62.16	57.07	33.65	36.32	39.67	52.82	46.95
GS-ICP-SLAM	Acc. ↓	04.80	10.96	25.10	06.93	05.83	09.00	10.44
	Comp. ↓	03.91	03.72	04.42	03.05	04.21	02.78	03.68
	Comp. Rat. ↑	86.31	83.73	76.94	86.99	75.04	89.11	83.02
	F-Score ↑	82.42	69.43	53.63	77.68	71.45	77.65	72.04
CaRtGS	Acc. ↓	09.19	13.78	27.49	11.85	19.65	14.62	16.10
	Comp. ↓	05.73	08.92	08.65	08.22	15.19	08.81	09.25
	Comp. Rat. ↑	63.84	46.23	45.29	45.71	26.31	47.55	45.82
	F-Score ↑	51.19	37.47	31.18	37.24	22.55	39.02	36.44
Ours	Acc. ↓	02.70	10.14	25.25	07.29	04.01	08.80	09.70
	Comp. ↓	02.11	02.88	03.03	02.31	02.22	02.28	02.47
	Comp. Rat. ↑	95.71	87.70	87.56	91.33	92.42	92.71	91.24
	F-Score ↑	93.94	70.26	60.16	79.33	88.16	82.75	79.10

D. Semantics: Zero-shot Segmentation

Beyond ensuring high-fidelity geometry and appearance reconstruction, OmniMap preserves instance-level scene understanding through its voxel representations \mathcal{V} and embedded codebook \mathcal{B} . We evaluate OmniMap against various offline and online open-vocabulary approaches by assessing their zero-shot semantic segmentation performance. Using dataset-provided semantic labels as query texts, we assign each map element the label with maximum embedding similarity, formulating the evaluation as a 3D multi-class classification task.

Tab. V and Tab. VI summarize the qualitative results of semantic segmentation on the two datasets. OmniMap outperforms all baselines, owing to its unique probabilistic voxel representation, which enhances robustness to noise. For visualization purposes, we assign distinct colors to each semantic label and display the segmentation results in Fig. 10. Built upon the YOLO-World model [58], OmniMap specifically focuses on foreground object segmentation, which is particularly valuable for robotic applications. The qualitative comparison highlights OmniMap’s advantage in semantic understanding. Point-wise feature fields like LangSplat struggle to distinguish between instances, while ConceptGraph and HOV-SG, though incorporating instance notions, suffer from redundant under-segmentation due to simple point-cloud IoU-based instance associations. In contrast, OmniMap achieves nearly identical results to the GT, demonstrating competitive performance in

TABLE V: Zero-shot semantic evaluations on Replica dataset.

Method	Metric	ro-0	ro-1	ro-2	of-0	of-1	of-2	of-3	of-4	Avg.
Open Gaussian	mIoU ↑	11.64	03.82	09.04	06.06	04.97	02.10	05.55	11.36	06.82
	fIoU ↑	31.07	10.44	22.17	10.14	15.32	03.15	07.00	24.00	15.41
	mAcc ↑	29.39	11.13	19.67	10.07	20.36	07.01	11.45	24.16	16.66
	fAcc ↑	05.96	14.35	28.09	12.24	16.90	04.92	07.90	24.31	18.08
Lang Splat	mIoU ↑	12.62	11.30	10.48	10.40	05.38	10.33	09.55	09.91	10.00
	fIoU ↑	47.83	47.74	41.56	29.69	24.49	54.13	44.79	27.29	39.69
	mAcc ↑	26.76	28.71	25.09	21.85	14.73	22.71	19.90	23.68	22.93
	fAcc ↑	53.77	53.70	50.51	31.94	28.57	56.78	50.24	27.77	44.16
Grasp Splats	mIoU ↑	16.74	08.25	14.13	11.35	02.77	11.13	14.79	04.17	10.42
	fIoU ↑	55.05	30.75	48.56	51.38	14.12	45.22	45.37	50.88	42.67
	mAcc ↑	32.92	21.74	32.45	33.51	08.89	22.12	27.92	10.80	23.79
	fAcc ↑	66.82	45.24	65.38	59.56	22.42	52.92	54.60	52.13	52.39
Concept Fusion	mIoU ↑	07.96	08.86	02.57	04.66	03.82	02.93	03.53	03.68	04.75
	fIoU ↑	32.28	34.86	31.65	23.45	20.35	23.35	19.15	17.29	25.30
	mAcc ↑	25.41	31.04	07.59	17.42	23.38	12.13	16.65	20.71	19.29
	fAcc ↑	38.11	42.19	37.44	24.86	22.82	24.43	21.85	20.23	28.99
Concept Graph	mIoU ↑	22.53	18.74	14.66	19.36	11.22	15.75	11.66	17.74	16.46
	fIoU ↑	50.28	46.21	54.52	43.87	22.00	23.25	11.32	34.07	35.69
	mAcc ↑	38.88	36.57	25.10	29.32	22.54	33.65	28.43	37.56	31.51
	fAcc ↑	59.65	57.93	62.64	50.21	25.64	27.78	16.05	39.63	42.44
HOV-SG	mIoU ↑	36.26	29.00	24.13	19.14	22.69	23.88	15.28	19.89	23.79
	fIoU ↑	71.24	51.95	62.52	44.05	49.50	46.28	20.95	44.42	48.86
	mAcc ↑	49.68	52.96	39.79	30.60	40.28	35.16	28.86	39.36	39.59
	fAcc ↑	81.69	61.36	72.31	46.72	59.72	47.81	24.90	46.71	55.15
Open Fusion	mIoU ↑	22.17	16.64	21.94	08.95	12.78	12.36	18.10	18.00	16.37
	fIoU ↑	60.17	41.62	61.72	49.64	33.86	48.22	53.00	64.93	51.65
	mAcc ↑	41.65	41.39	43.03	25.61	30.01	26.99	32.14	40.40	35.15
	fAcc ↑	72.78	54.68	73.71	60.10	37.01	52.82	62.53	69.33	60.37
Ours	mIoU ↑	49.57	42.63	26.50	21.92	21.32	24.45	23.31	22.82	29.06
	fIoU ↑	70.73	57.89	65.56	62.37	52.14	76.53	62.30	67.85	64.42
	mAcc ↑	62.94	61.68	41.04	38.28	32.76	39.42	37.17	43.03	44.54
	fAcc ↑	82.29	69.93	74.66	71.40	55.29	81.94	70.85	71.37	72.22

both segmentation and classification tasks.

E. Efficiency: Frame Rate and Model Size

For real-time applications across diverse robotic platforms, frame rate and model size serve as critical metrics for evaluating algorithm performance. While theoretically any method could achieve improved results through excessive Gaussian primitives (increasing model size) or extended per-frame optimization (reducing frame rate), such approaches prove impractical for real-world deployment. To verify that OmniMap maintains efficient construction while delivering high-quality reconstructions, we measure both build frame rate and model size in Tab. VII, alongside rendering and reconstruction metrics for reference. Experimental results demonstrate that OmniMap achieves an optimal balance between real-time performance (FPS), model compactness, and reconstruction quality compared to all baseline methods. This confirms OmniMap’s ability to maintain high mapping performance without compromising efficiency.

F. Ablation Study

To evaluate the effectiveness of the proposed components, we conducted an ablation study for ScanNet’s scene0181_00 and report the results in Tab. VIII. The significance of probabilistic voxel representation for open-vocabulary semantics has been established in Subsection VII-D. We further evaluate

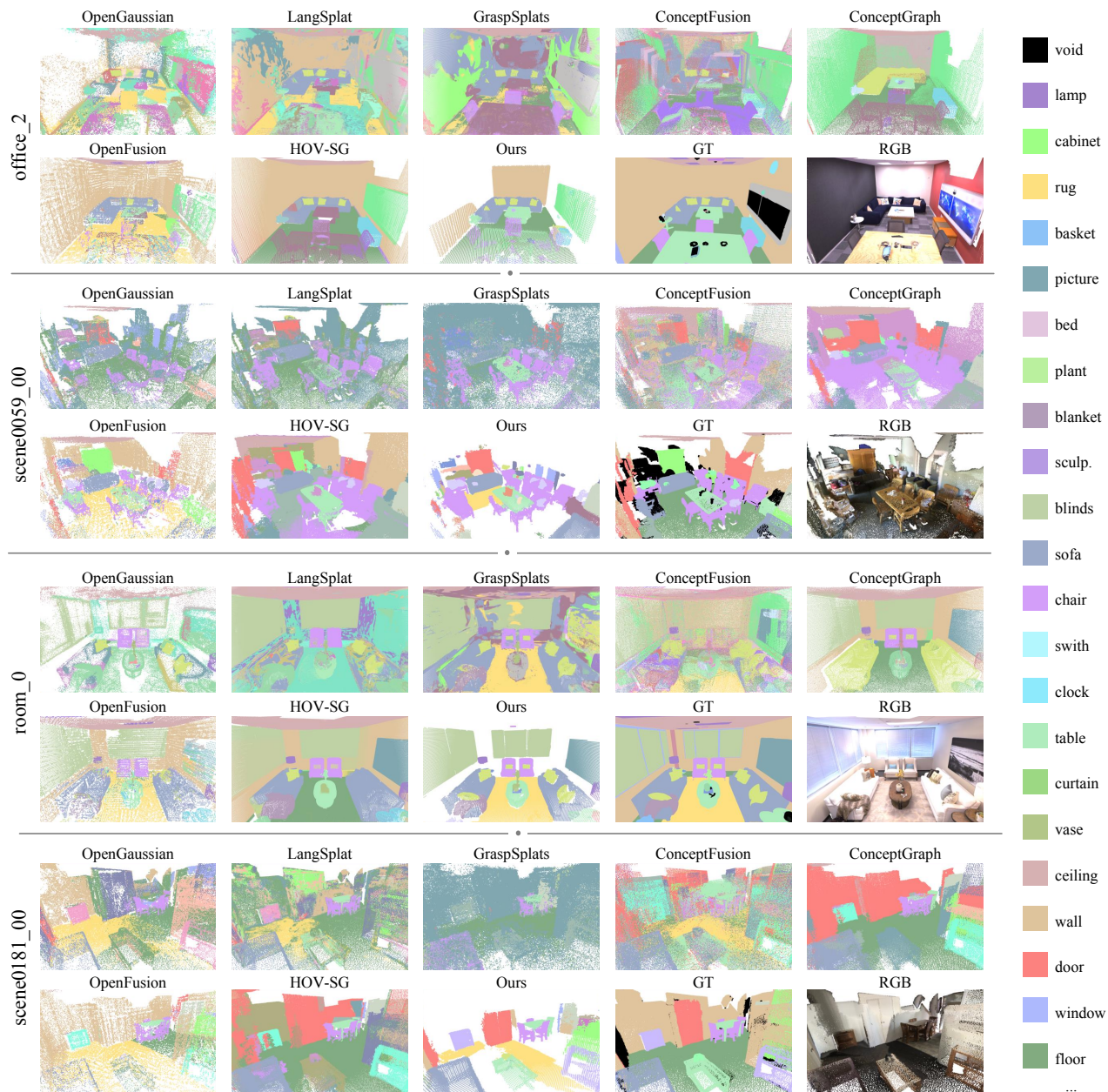


Fig. 10: Zero-shot semantic segmentation results. OmniMap’s probabilistic voxel representation demonstrates robustness to sensor noise while maintaining accurate object segmentation and scene understanding capabilities.

three additional key modules’ impact on rendering and reconstruction. Our quantitative analysis demonstrates that: (1) removal of the camera parameter modeling module leads to significant degradation in rendering quality, particularly in uncertain real-world environments; (2) while normal supervision removal has minimal impact on quantitative rendering metrics, it substantially reduces mesh accuracy, with qualitative results showing pronounced deterioration (see Fig. 7); and (3) elimination of the keyframe buffer results in noticeable declines in both optical and geometric performance due to catastrophic forgetting effects.

We further evaluate the impact of two key parameters on mapping performance: voxel resolution and instance fusion threshold. Voxel resolution directly determines the number of

Gaussian primitives, influencing both rendering quality and mesh accuracy, while the instance fusion threshold governs instance segmentation granularity. As shown in Fig. 11, increasing the voxel size reduces model complexity (through fewer Gaussian primitives) while marginally decreasing optical and geometric accuracy, enabling a practical accuracy-efficiency trade-off. Notably, OmniMap with a voxel resolution of 0.06m achieves minimal model size (6.2 MB only), while still maintaining the highest rendering and mesh accuracy. Although higher fusion thresholds affect semantic accuracy to some degree, the impact remains within acceptable bounds, demonstrating OmniMap’s robustness to parameter variations in probabilistic modeling. The bottom row of Fig. 11 illustrates fusion results at varying thresholds, highlighting differences in

TABLE VI: Zero-shot semantic evaluations on ScanNet dataset.

Method	Metric	0000	0059	0106	0169	0181	0207	Avg.
Open Gaussian	mIoU↑	10.84	05.83	09.08	09.29	11.38	05.44	08.64
	fIoU↑	18.52	27.51	15.67	54.55	19.59	06.44	23.71
	mAcc↑	21.82	12.75	22.55	17.15	24.41	08.50	17.86
	fAcc↑	22.68	30.08	17.92	57.10	23.59	07.25	26.44
Lang Splat	mIoU↑	06.72	07.41	08.55	06.02	11.13	03.48	07.22
	fIoU↑	20.03	46.98	29.48	43.60	23.06	02.38	27.59
	mAcc↑	18.99	22.48	25.14	25.39	28.87	05.18	21.01
	fAcc↑	26.26	52.80	36.09	47.34	28.21	02.54	32.21
Grasp Splats	mIoU↑	06.27	08.22	06.46	08.40	05.53	05.27	06.69
	fIoU↑	08.39	25.88	19.26	30.77	07.58	07.82	16.62
	mAcc↑	12.27	20.24	09.46	26.28	10.80	10.29	14.89
	fAcc↑	09.60	28.87	21.12	38.66	09.79	09.40	19.57
Concept Fusion	mIoU↑	11.13	05.48	09.55	06.15	09.92	08.77	08.50
	fIoU↑	23.70	32.18	37.65	42.38	29.15	15.24	30.05
	mAcc↑	34.92	28.69	37.16	30.35	35.65	24.06	31.81
	fAcc↑	31.50	37.36	45.10	47.92	36.52	21.41	36.64
Concept Graph	mIoU↑	15.38	16.06	18.57	17.78	16.55	13.42	16.29
	fIoU↑	16.17	42.06	39.11	52.52	30.56	19.31	33.29
	mAcc↑	35.69	32.87	40.52	37.67	32.09	25.60	34.07
	fAcc↑	21.84	53.36	47.93	64.08	38.27	24.15	41.60
HOV-SG	mIoU↑	24.57	18.19	22.90	25.49	16.52	16.92	20.76
	fIoU↑	31.93	42.53	37.66	58.84	28.58	30.51	38.34
	mAcc↑	46.78	37.98	47.41	45.21	37.08	34.56	41.50
	fAcc↑	38.74	51.73	42.41	69.97	33.47	36.65	45.50
Open Fusion	mIoU↑	18.58	12.09	15.76	19.32	21.26	21.11	18.02
	fIoU↑	28.56	48.04	36.22	59.03	50.44	40.60	43.82
	mAcc↑	47.85	44.21	42.95	39.06	43.27	48.52	44.31
	fAcc↑	38.84	56.89	39.51	62.63	57.10	51.56	51.09
Ours	mIoU↑	20.25	14.81	38.44	24.41	33.04	21.59	25.42
	fIoU↑	34.24	53.59	64.18	57.34	54.96	40.86	50.86
	mAcc↑	49.81	47.36	62.37	41.43	61.61	42.99	50.93
	fAcc↑	40.95	61.26	71.72	62.77	59.90	45.68	57.05

TABLE VII: Comparisons of mapping efficiency and model size.

Dataset	Replica			Scannet				
	Metric	PSNR↑	FPS↑	Size↓	PSNR↑	F-Score↑	FPS↑	Size↓
RTG-SLAM	32.21	2.34	163.4	18.11	67.93	0.91	460.0	
Gaussian-SLAM	36.37	1.93	81.4	23.35	60.24	2.35	59.8	
MonoGS	36.23	0.73	10.5	18.88	72.20	0.90	15.2	
SplaTAM	34.81	0.30	58.1	19.09	76.42	0.47	181.5	
GSFusion	34.70	3.91	36.2	19.03	46.95	5.22	86.7	
GS-ICP-SLAM	37.47	6.56	21.6	20.34	72.04	7.01	25.0	
CaRtGS	34.88	5.24	23.7	19.12	36.44	9.98	18.2	
HI-SLAM2	36.59	5.68	11.6	23.37	-	3.00	10.6	
Ours	39.48	5.43	14.2	26.04	79.10	5.55	40.5	

TABLE VIII: Results of ablation study.

Metric	PSNR↑	F-Score↑
w/o camera modeling	25.07	86.81
w/o normal loss	26.29	84.56
w/o keyframe buffer	21.16	86.46
Full OmniMap	26.33	88.16

challenging cases such as box bases on couches, adjacent trash cans, and highly overlapping tables and chairs.

VIII. APPLICATIONS

OmniMap’s unified representation of optical, geometric, and semantic information provides inherent advantages for supporting diverse downstream applications. This integrated framework naturally enables multiple use cases, including (1)

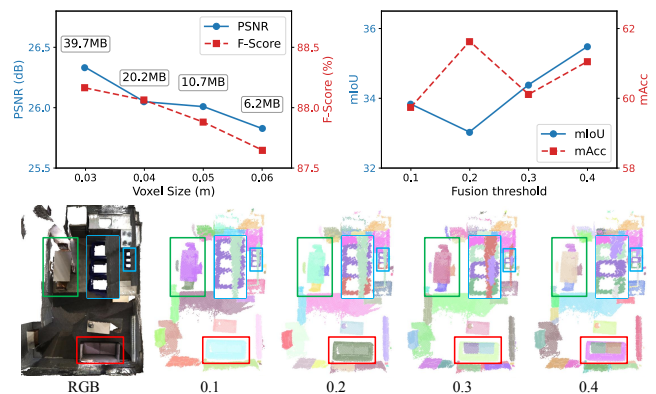


Fig. 11: The impact of two key parameters on mapping performance: voxel resolution and instance fusion threshold. The bottom row shows the final instance-level maps for different fusion thresholds.

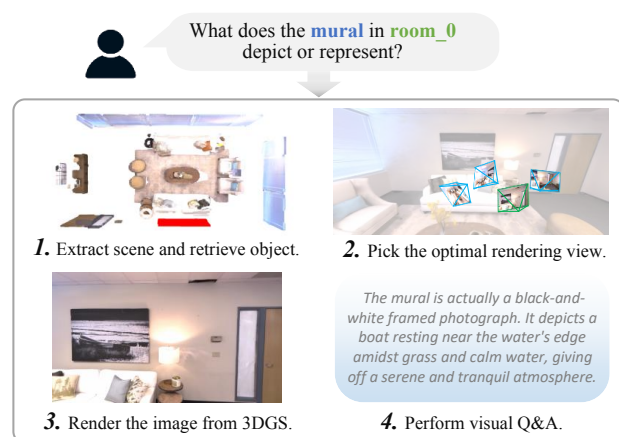


Fig. 12: The process of 2D Scene Q&A. OmniMap enables scene understanding and question answering through object retrieval and image rendering capabilities, augmented by VQA techniques.

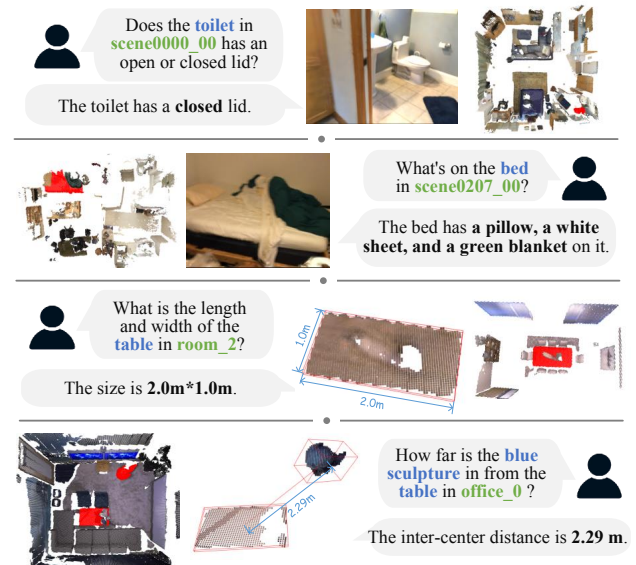


Fig. 13: More demonstrations of Scene Q&A.

Scene Q&A, (2) Interactive Editing, (3) Perception-guided Manipulation, and (4) Map-assisted Navigation.

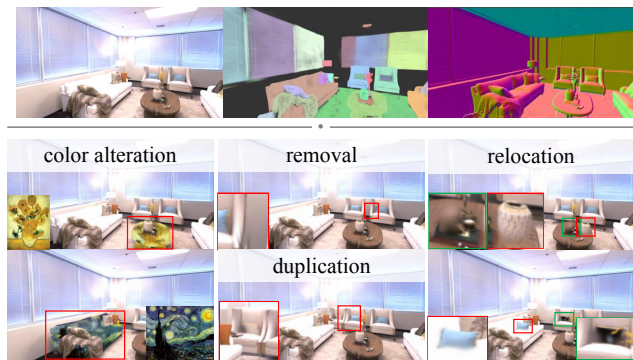


Fig. 14: Scene editing demonstration. Through object querying and direct manipulation of Gaussian primitives, our framework enables flexible scene modification with multi-view rendering capabilities.

A. Scene Q&A

After completing environmental reconstruction, scene quizzing serves as an effective tool for validating representation quality. User queries typically target specific object-level information, demanding rapid localization of relevant objects and extraction of fine-grained attributes like appearance. Fig. 12 demonstrates OmniMap’s Scene Q&A pipeline with the following workflow:

1. Object retrieval: Leveraging the embedded codebook, we employ feature similarity matching to efficiently retrieve the most relevant object ID and localize its corresponding voxels.

2. Viewpoint selection: As visual perception dominates scene understanding, we project the identified voxels across multiple camera poses. By comparing projection depth with rendering depth, we select the optimal viewpoint maximizing valid voxel projections.

3. Image rendering: The 3DGS field is rendered at the chosen viewpoint to generate RGB and optional depth images (for 3D-related queries).

4. Visual Q&A: Utilizing Multimodal Large Language Models (MLLMs) capabilities, we provide comprehensive responses to user queries. An optional feedback mechanism dynamically adjusts rendering viewpoints during Q&A to enhance visual representation.

Beyond basic 2D scene Q&A, OmniMap inherently supports 3D queries, enabling object dimension measurement and inter-object distance calculation. Fig. 13 demonstrates additional scene query results encompassing both 2D and 3D capabilities.

B. Interactive Editing

Thanks to OmniMap’s instance-level scene decomposition and semantic understanding, we enable intuitive interactive scene editing with free-viewpoint rendering. Specifically, given an object-level editing command, we first retrieve the target object’s voxels. These voxels are then filtered by a distance threshold and passed to the 3D Gaussian Splatting (3DGS) field to extract the associated Gaussian primitives. By directly manipulating these primitives, the modified Gaussian field can be rendered from any viewpoint. Supported edits include

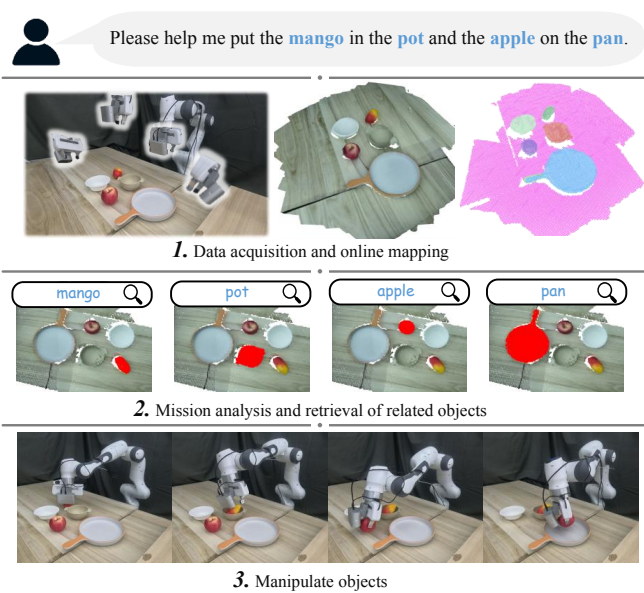


Fig. 15: OmniMap supports robotic arm operations by reconstructing workspace optics, geometry, and semantics through end-effector-mounted camera scanning, enabling task completion.

color alteration, removal, duplication, and relocation, as illustrated in Fig. 14. This interactive editing capability allows OmniMap to efficiently generate diverse scene layouts and visual appearances while maintaining high-fidelity rendering quality, offering an efficient approach to address critical data augmentation challenges in embodied intelligence research.

C. Perception-guided Manipulation

Beyond interactive mapping applications, OmniMap demonstrates direct hardware control capabilities for robotic systems. We first validated this through robotic arm manipulation experiments. As illustrated in Fig. 15, an Orbbec Astra Pro camera was mounted on the end-effector of a Franka robotic arm. When presented with the user command ‘Please help me put the mango in the pot and the apple on the pan’, the system executed a comprehensive tabletop scanning procedure: the robotic arm maneuvered the camera around the workspace, enabling OmniMap to perform rapid incremental reconstruction. The camera pose was precisely determined through end-effector tracking combined with calibrated extrinsic parameters.

This process yielded not only a complete geometric map but also a semantic understanding of the objects within the scene. OmniMap then localized the task-relevant objects (mango, pot, apple, and pan) within the reconstructed environment. Leveraging LLM-based task decomposition, the system successfully guided the robotic arm through sequential operations to complete the requested tasks with precise object manipulation.

D. Map-assisted Navigation

We conducted real-world validation of OmniMap’s effectiveness in an office environment using both a wheeled robot and a quadruped robotic platform. The wheeled robot,

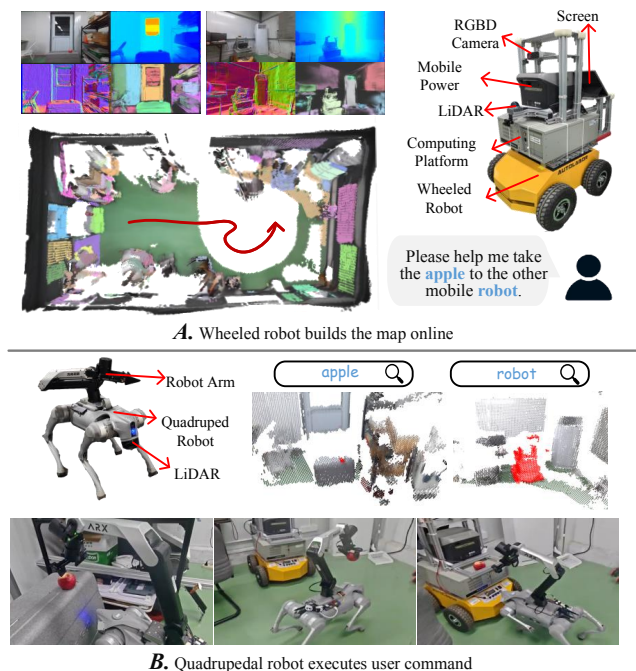


Fig. 16: OmniMap enables mobile robot navigation experiments. A wheeled robot performs environmental scanning to generate comprehensive scene reconstruction and semantic understanding, while a quadrupedal robot with a manipulator utilizes this representation to complete user-specified tasks within the mapped environment.

equipped with an Azure Kinect RGB-D camera and an on-board computing system (featuring an RTX 3090 GPU), utilized Livox MID-360 LiDAR SLAM along with multi-sensor calibration technology to generate images' poses while simultaneously streaming RGB-D data to OmniMap for processing.

OmniMap successfully performs online environment reconstruction as the robot traverses the scene. As illustrated in Fig. 16, the system demonstrates robust real-time mapping performance and produces high-quality reconstruction results. Notably, despite challenging conditions including ground surface reflections, non-uniform lighting distribution, and cluttered objects, OmniMap maintains remarkable reconstruction fidelity, with its real-time rendered output closely matching the actual RGB feed. This achievement highlights the system's exceptional perceptual capabilities. Furthermore, the randomly color-coded instance mapping results provide compelling evidence of OmniMap's superior performance in real-world applications involving simultaneous instance detection, semantic segmentation, and environmental reconstruction tasks.

To validate the practical deployment of the reconstructed map, we implemented it on a quadrupedal robot equipped with a 6-DOF manipulator. As shown in Fig. 16, when given the natural language command 'Please help me take the apple to the other mobile robot', the system successfully identified and localized both target objects, the apple and the robot, through LLM-based semantic parsing. The robot then autonomously planned and executed a collision-free trajectory using the generated obstacle map, demonstrating precise navigation and manipulation capabilities as it grasped and transported the apple to the specified target location. This complete functional

chain from perception to action conclusively verifies the operational readiness of our mapping system for real-world service robotics applications.

IX. CONCLUSION

In this work, we present OmniMap, a general framework for online mapping that simultaneously incorporates optical (high-fidelity rendering), geometric (fine-grained mesh), and semantic (open-vocabulary understanding) representations. The framework introduces several key innovations, including hybrid 3DGS-Voxel architecture, differentiable camera parameter modeling, voxel-based Gaussian initialization, higher-order geometric signal supervision, and probabilistic instance-level reconstruction. These technical advances enable OmniMap to achieve comprehensive optimization across all performance metrics in nearly all scenes. Efficiency evaluations demonstrate that OmniMap accomplishes these results without compromising computational efficiency or memory usage, with ablation studies confirming that even with significantly reduced model sizes, the system maintains high accuracy. Furthermore, the unified nature of OmniMap supports diverse downstream applications such as scene-based multi-domain question answering, interactive scene editing, robotic arm manipulation, and mobile robot navigation.

However, OmniMap currently has limitations: its mapping quality depends on external positioning systems due to the lack of an integrated tracking module, and it cannot effectively handle dynamic scenes. Future work will focus on developing an integrated tracking module and implementing dynamic instance discrimination to transform OmniMap into a more versatile, plug-and-play scene representation solution.

REFERENCES

- [1] A. Hornung, K. M. Wurm, M. Bennewitz, *et al.*, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [2] H. Oleynikova, Z. Taylor, M. Fehr, *et al.*, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1366–1373. IEEE, 2017.
- [3] R. A. Newcombe, S. Izadi, O. Hilliges, *et al.*, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*, pp. 127–136. Ieee, 2011.
- [4] A. Dai, M. Nießner, M. Zollhöfer, *et al.*, "Bundlfusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.
- [5] B. Mildenhall, P. P. Srinivasan, M. Tancik, *et al.*, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [6] B. Kerbl, G. Kopanas, T. Leimkühler, *et al.*, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.
- [7] Q. Gu, A. Kuwajerwala, S. Morin, *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *IEEE international conference on robotics and automation (ICRA)*. IEEE, 2024.
- [8] A. Werby, C. Huang, M. Büchner, *et al.*, "Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, Jul. 2024.
- [9] J. Kerr, C. M. Kim, K. Goldberg, *et al.*, "Lerf: Language embedded radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19 729–19 739, 2023.
- [10] M. Qin, W. Li, J. Zhou, *et al.*, "Langsplat: 3d language gaussian splatting," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

- [11] Y. Cai, F. Kong, Y. Ren, *et al.*, “Occupancy grid mapping without ray-casting for high-resolution lidar sensors,” *IEEE Transactions on Robotics*, vol. 40, pp. 172–192, 2023.
- [12] G. Chen, W. Dong, P. Peng, *et al.*, “Continuous occupancy mapping in dynamic environments using particles,” *IEEE Transactions on Robotics*, vol. 40, pp. 64–84, 2023.
- [13] Y. Yuan and A. Nüchter, “Uni-fusion: Universal continuous mapping,” *IEEE Transactions on Robotics*, vol. 40, pp. 1373–1392, 2024.
- [14] Y. Deng, M. Wang, Y. Yang, *et al.*, “See-csom: Sharp-edged and efficient continuous semantic occupancy mapping for mobile robots,” *IEEE Transactions on Industrial Electronics*, 2023.
- [15] A. Asgharivaskasi and N. Atanasov, “Semantic octree mapping and shannon mutual information computation for robot exploration,” *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1910–1928, 2023.
- [16] Y. Deng, M. Wang, Y. Yang, *et al.*, “Hd-ccsom: Hierarchical and dense collaborative continuous semantic occupancy mapping through label diffusion,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2417–2422. IEEE, 2022.
- [17] V. Yugay, Y. Li, T. Gevers, *et al.*, “Gaussian-slam: Photo-realistic dense slam with gaussian splatting,” *arXiv preprint arXiv:2312.10070*, 2023.
- [18] N. Keetha, J. Karhade, K. M. Jatavallabhula, *et al.*, “Splatam: Splat track & map 3d gaussians for dense rgb-d slam,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21 357–21 366, 2024.
- [19] H. Matsuki, R. Murai, P. H. Kelly, *et al.*, “Gaussian splatting slam,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18 039–18 048, 2024.
- [20] K. Jatavallabhula, A. Kuwajerwala, Q. Gu, *et al.*, “Conceptfusion: Open-set multimodal 3d mapping,” *Robotics: Science and Systems (RSS)*, 2023.
- [21] Y. Deng, J. Wang, J. Zhao, *et al.*, “Opengraph: Open-vocabulary hierarchical 3d graph representation in large-scale outdoor environments,” *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8402–8409, 2024.
- [22] J. Wei and S. Leutenegger, “Gsfusion: Online rgb-d mapping where gaussian splatting meets tsdf fusion,” *IEEE Robotics and Automation Letters*, 2024.
- [23] Z. Zhu, S. Peng, V. Larsson, *et al.*, “Nice-slam: Neural implicit scalable encoding for slam,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12 786–12 796, 2022.
- [24] L. Jin, X. Zhong, Y. Pan, *et al.*, “Activegs: Active scene reconstruction using gaussian splatting,” *IEEE Robotics and Automation Letters*, 2025.
- [25] L. Ma, X. Li, J. Liao, *et al.*, “Deblur-nerf: Neural radiance fields from blurry images,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12 861–12 870, 2022.
- [26] B. Lee, H. Lee, X. Sun, *et al.*, “Deblurring 3d gaussian splatting,” in *European Conference on Computer Vision*, pp. 127–143. Springer, 2024.
- [27] L. Zhao, P. Wang, and P. Liu, “Bad-gaussians: Bundle adjusted deblur gaussian splatting,” in *European Conference on Computer Vision*, pp. 233–250. Springer, 2024.
- [28] K. Yamazaki, T. Hanyu, K. Vo, *et al.*, “Open-fusion: Real-time open-vocabulary 3d mapping and queryable scene representation,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9411–9417. IEEE, 2024.
- [29] D. Meyer-Delius, M. Beinhofer, and W. Burgard, “Occupancy grid models for robot mapping in changing environments,” in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [30] L. Han, F. Gao, B. Zhou, *et al.*, “Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4423–4430. IEEE, 2019.
- [31] Y. Xiang and D. Fox, “Da-rnn: Semantic mapping with data associated recurrent neural networks,” in *Robotics: Science and Systems (RSS)*, 2017.
- [32] J. McCormac, A. Handa, A. Davison, *et al.*, “Semanticfusion: Dense 3d semantic mapping with convolutional neural networks,” in *2017 IEEE International Conference on Robotics and automation (ICRA)*, pp. 4628–4635. IEEE, 2017.
- [33] S. Kim and J. Kim, “Gpmap: A unified framework for robotic mapping based on sparse gaussian processes,” in *Field and Service Robotics: Results of the 9th International Conference*, pp. 319–332. Springer, 2015.
- [34] M. G. Jadidi, L. Gan, S. A. Parkison, *et al.*, “Gaussian processes semantic map representation,” *arXiv preprint arXiv:1707.01532*, 2017.
- [35] F. Ramos and L. Ott, “Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent,” *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1717–1730, 2016.
- [36] T. Shan, J. Wang, B. Englot, *et al.*, “Bayesian generalized kernel inference for terrain traversability mapping,” in *Conference on Robot Learning*, pp. 829–838. PMLR, 2018.
- [37] L. Gan, Y. Kim, J. W. Grizzle, *et al.*, “Multitask learning for scalable and dense multilayer bayesian map inference,” *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 699–717, 2022.
- [38] Y. Deng, M. Wang, D. Wang, *et al.*, “S-mki: Incremental dense semantic occupancy reconstruction through multi-entropy kernel inference,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3824–3829. IEEE, 2022.
- [39] J. J. Park, P. Florence, J. Straub, *et al.*, “Deepsdf: Learning continuous signed distance functions for shape representation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019.
- [40] L. Mescheder, M. Oechsle, M. Niemeyer, *et al.*, “Occupancy networks: Learning 3d reconstruction in function space,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4460–4470, 2019.
- [41] K. Genova, F. Cole, A. Sud, *et al.*, “Local deep implicit functions for 3d shape,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4857–4866, 2020.
- [42] Y. Yue, Y. Deng, Y. Tang, *et al.*, “Lgdsf: Continual global learning of signed distance fields aided by local updating,” *IEEE Robotics and Automation Letters*, 2025.
- [43] Y. Deng, Y. Tang, Y. Yang, *et al.*, “Macim: Multi-agent collaborative implicit mapping,” *IEEE Robotics and Automation Letters*, 2024.
- [44] E. Sucar, S. Liu, J. Ortiz, *et al.*, “imap: Implicit mapping and positioning in real-time,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6229–6238, 2021.
- [45] V. Sitzmann, S. Rezkchikov, B. Freeman, *et al.*, “Light field networks: Neural scene representations with single-evaluation rendering,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 19 313–19 325, 2021.
- [46] T. Müller, A. Evans, C. Schied, *et al.*, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.
- [47] A. Chen, Z. Xu, A. Geiger, *et al.*, “Tensorf: Tensorial radiance fields,” in *European conference on computer vision*, pp. 333–350. Springer, 2022.
- [48] C. Yan, D. Qu, D. Xu, *et al.*, “Gs-slam: Dense visual slam with 3d gaussian splatting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19 595–19 604, 2024.
- [49] Z. Peng, T. Shao, Y. Liu, *et al.*, “Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting,” in *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.
- [50] E. Sandström, G. Zhang, K. Tateno, *et al.*, “Splat-slam: Globally optimized rgb-only slam with 3d gaussians,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1680–1691, 2025.
- [51] S. Ha, J. Yeon, and H. Yu, “Rgbd gs-icp slam,” in *European Conference on Computer Vision*, pp. 180–197. Springer, 2024.
- [52] D. Feng, Z. Chen, Y. Yin, *et al.*, “Cartgs: Computational alignment for real-time gaussian splatting slam,” *IEEE Robotics and Automation Letters*, 2025.
- [53] W. Zhang, Q. Cheng, D. Skuddis, *et al.*, “Hi-slam2: Geometry-aware gaussian slam for fast monocular scene reconstruction,” *arXiv preprint arXiv:2411.17982*, 2024.
- [54] S. Kobayashi, E. Matsumoto, and V. Sitzmann, “Decomposing nerf for editing via feature field distillation,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 311–23 330, 2022.
- [55] Y. Deng, J. Wang, J. Zhao, *et al.*, “Openobj: Open-vocabulary object-level neural radiance fields with fine-grained understanding,” *IEEE Robotics and Automation Letters*, 2024.
- [56] M. Ye, M. Danelljan, F. Yu, *et al.*, “Gaussian grouping: Segment and edit anything in 3d scenes,” in *European Conference on Computer Vision*, pp. 162–179. Springer, 2024.
- [57] Y. Wu, J. Meng, H. Li, *et al.*, “Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 19 114–19 138, 2024.
- [58] T. Cheng, L. Song, Y. Ge, *et al.*, “Yolo-world: Real-time open-vocabulary object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16 901–16 911, 2024.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

- [59] T. Pan, L. Tang, X. Wang, *et al.*, “Tokenize anything via prompting,” in *Proceedings of the European conference on computer vision (ECCV)*, 2024.
- [60] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 11 2019.
- [61] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 303–312, 1996.
- [62] L. Gan, R. Zhang, J. W. Grizzle, *et al.*, “Bayesian spatial kernel smoothing for scalable dense semantic mapping,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 790–797, 2020.
- [63] M. Ji, R.-Z. Qiu, X. Zou, *et al.*, “Graspsplats: Efficient manipulation with 3d feature splatting,” in *Conference on Robot Learning*, 2024.
- [64] J. Straub, T. Whelan, L. Ma, *et al.*, “The replica dataset: A digital replica of indoor spaces,” *arXiv preprint arXiv:1906.05797*, 2019.
- [65] A. Dai, A. X. Chang, M. Savva, *et al.*, “ScanNet: Richly-annotated 3d reconstructions of indoor scenes,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.