IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Memory-Efficient Voxelized Renderable Neural 3D Spatial Representation for Vision-Based Robotics

Howoong Jun[1], Seongbo Ha[2], Jaewon Lee[3], Hyeonwoo Yu[2], and Songhwai Oh[4]

***Abstract*—In this paper, we introduce a novel approach for modeling a memory-efficient spatial representation with 3D Gaussian splatting. Efficient vision-based spatial representation poses a significant challenge due to the memory demands of visual information. Recent advances in 3D rendering technologies, such as neural radiance fields (NeRF) and 3D Gaussian splatting, have prompted exploration of their applications in robotics. However, such 3D rendering methods often focus on rendering high-quality images, requiring numerous parameters and resulting in large data, which are unsuitable for robotics applications. To tackle this challenge, we introduce 3DSR, an efficient voxelized renderable neural 3D spatial representation that utilizes 3D Gaussian splatting. 3DSR leverages the strengths of both voxelization (memory efficiency) and 3D Gaussian splatting (high-quality image reconstruction). The proposed method achieves memory efficiency by reducing the number of 3D Gaussians in the 3D representation through voxelization, while preserving the image quality required for effective vision-based robotic applications. Experimental results demonstrate that 3DSR achieves over 90% of the best method's reconstruction quality while requiring only 54.54% of its memory. Additional experiments on visual localization and navigation further confirm that the proposed method is readily applicable to robotics.**

***Index Terms*—3D Gaussian splatting, efficient spatial representation, vision-based robotics.**

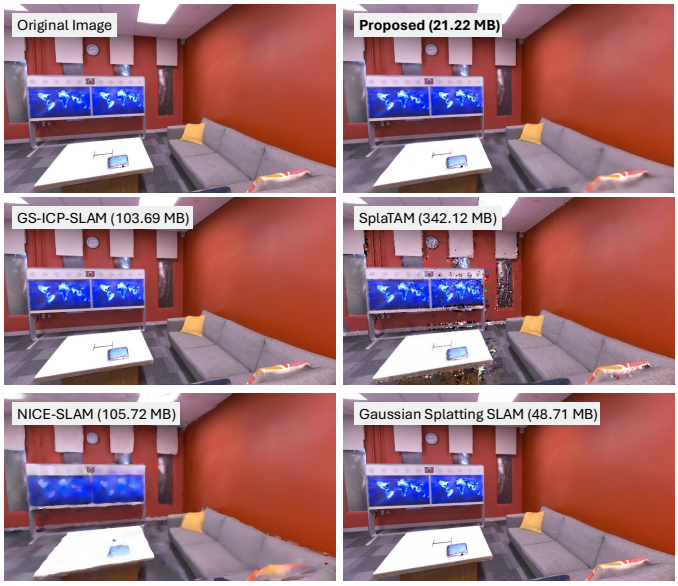

Fig. 1. Overview of the results obtained from the proposed method. Qualitative evaluation demonstrates that the proposed method minimizes memory consumption while producing renderings that closely match the original image.

I. Introduction

SPATIAL representation is a fundamental capability for autonomous robots, enabling them to localize, perceive, and navigate through complex environments. With the rise of vision-based robotics, two key challenges emerge in this field: (1) efficient spatial representation of visual information and (2) dense spatial representation. Efficient spatial representation is crucial, as robots must process large amounts of information while operating in real time. Additionally, robots require well-distributed spatial representations that accurately capture the physical worlds, as gaps or holes can lead to accidents. One mainstream approach to overcome the issue involves using 3D occupancy maps, which aggregate point clouds into voxels [1], [2] or directly estimate 3D occupancy grids through multiple cameras [3], [4]. This 3D voxel representation is widely adopted in robotics due to its intuitiveness, compatibility with various sensors, and efficiency in representing physical worlds. While these approaches are widely used in robotics, they often compromise the detailed visual information essential for robots to perceive the environment.

Recently, advanced neural field-based 3D rendering technologies, such as neural radiance fields (NeRF) [5] and 3D Gaussian splatting [6], have sparked interest among robotics researchers due to their ability to represent any 3D scenes with high fidelity [7]. These methods have earned significant attention in the field of simultaneous localization and mapping (SLAM) due to their ability to simultaneously capture 3D geometry and generate high-quality 2D images [8]–[13]. While these approaches offer optimized representations of visual information of the 3D space, they prioritize visual fidelity over

Manuscript received: July, 5, 2025; Revised September, 12, 2025; Accepted October 20, 2025. This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2019-II191190, [SW Star Lab] Robot Learning: Efficient, Safe, and Socially-Acceptable Machine Learning, 50%) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2022-NR070264, General-Purpose Deep Reinforcement Learning Using Metaverse for Real World Applications, 50%). *(Corresponding author: Songhwai Oh.)*

[1]Howoong Jun is with Interdisciplinary Program in Artificial Intelligence, Seoul National University (SNU) and Automation and Systems Research Institute (ASRI), and Sequor Robotics, Seoul, Korea (Republic of). howoong.jun@rllab.snu.ac.kr

[2]Seongbo Ha and Hyeonwo Yu are with Department of Intelligent Robotics, Sungkyunkwan University (SKKU), Suwon-si, Gyeonggi-do, Korea (Republic of). sobo3607@g.skku.edu, hwyu@skku.edu

[3]Jaewon Lee is with the Department of Electrical and Computer Engineering, Seoul National University (SNU) and Automation and Systems Research Institute (ASRI), and Sequor Robotics, Seoul, Korea (Republic of). jaewon.lee@rllab.snu.ac.kr

[4]Songhwai Oh is with the Department of Electrical and Computer Engineering & Interdisciplinary Program in Artificial Intelligence, Seoul National University (SNU) and Automation and Systems Research Institute (ASRI), and Sequor Robotics, Seoul, Korea (Republic of). songhwai@snu.ac.kr

memory efficiency and structural representation of physical worlds, which are crucial for robotics.

In response to these challenges, new approaches have emerged that balance realistic scene representation with memory efficiency [14]–[16]. However, these methods primarily focus on object-centered 3D reconstruction rather than capturing large-scale 3D environments, limiting their applicability to robotics. Consequently, research specifically focused on optimizing the visual representation of large-scale 3D spaces for robotic applications, while maintaining efficiency, still remains limited.

To address this deficiency, we focus on efficient spatial representation for robotic applications using 3D Gaussian splatting in this work. In this paper, we propose a voxelized-renderable neural 3D spatial representation (3DSR) which integrates the advantages of voxel representation and 3D Gaussian splatting. Figure 1 describes the sample comparison results of the proposed method. It employs a voxel representation for memory efficiency while preserving meaningful visual information by incorporating 3D Gaussian splatting and an upsampling layer into the rendering process. Building on the initial dense 3D Gaussian splatting, the proposed method follows two steps: (step 1) voxelization and (step 2) upsampling of the rendered images.

We evaluate 3DSR on multiple datasets—Replica [17], TUM [18], and HM3D [19]—which provide diverse environments for validating image rendering quality and vision-based robotic applications. In terms of rendering performance, our method achieves over 90% of the reconstruction quality of the best-performing baseline while using only 54.54% of its memory. In addition, visual localization experiments show that the ability of 3DSR to render novel views enhances localization accuracy by an average of 36.59%. Additionally, visual navigation results validate the applicability of our method to data-driven robotics tasks, which typically require extensive data augmentation. Overall, the key contributions of this work are as follows:

- We introduce an efficient 3D Gaussian splatting representation for vision-based robotics that reduces memory overhead while maintaining high-fidelity rendering.
- The compact representation readily accommodates extra modalities (e.g., localization features) with minimal additional memory.
- The method also serves as a powerful data-augmentation tool for data-driven robotics applications.

II. Related Work

a) NeRF-Based Spatial Representation: iMAP [8] pioneers the integration of implicit neural scene representation into the SLAM problem. The method uses a multilayer perceptron (MLP) as the sole scene representation for RGB-D camera. NICE-SLAM [9] introduces a hierarchical, grid-based neural implicit encoding that enables local updates, making it suitable for large-scale scenes. The method demonstrates real-time capability, scalability, and robustness in various challenging scenarios. vMAP [10] intoduces object-level dense SLAM system that uses neural field representations to model individual objects in a scene. The system employs vectorized training to efficiently optimize multiple object models in parallel and achieve improved scene-level and object-level reconstruction quality compared to previous neural field SLAM systems.

b) 3D Gaussian Splatting-Based Spatial Representation: 3D Gaussian splatting is another popular method for spatial representation. Gaussian splatting SLAM [11] leverages the efficiency of 3D Gaussian splatting for scene representation and rendering, while incorporating a keyframe-based SLAM framework to achieve real-time performance and handle large-scale environments. SplaTAM [13] enhances the accuracy and efficiency of SLAM systems by leveraging the continuous representation of 3D Gaussians, which allows for better handling of sensor noise and dynamic environments. SplaTAM demonstrates significant improvements in both tracking and mapping performance compared to other baselines. GS-ICP-SLAM [12] introduces a novel dense representation SLAM approach that combines generalized iterative closest point (G-ICP) [20] and 3D Gaussian splatting techniques. This approach achieves both high-frequency tracking performance and high-quality map reconstruction.

III. Method

The primary objective of the 3DSR approach is to produce a memory-efficient 3D Gaussian splatting representation suitable for robotic applications. The method comprises two main steps: base 3D Gaussians creation and image upsampling. We assume that the original dense 3D Gaussian splatting representation is given beforehand. First, the 3D Gaussian splatting-based point clouds are downsampled through voxelization to significantly reduce memory usage. This voxelized representation, referred to as the 'base 3D Gaussians', serves as the foundation for rendering coarse images. When a 6-DoF pose is provided, a coarse image is rendered from the base 3D Gaussians. Subsequently, the coarse image is upsampled to its original resolution and quality using an upsampling network, which is trained offline. Figure 2 describes the overall process of the proposed 3DSR method.

A. Voxelized-Renderable Neural 3D Spatial Representation

a) Base 3D Gaussian Creation: Initially, we create an original 3D Gaussian splatting representation $\mathbb{M}_o = \{G_o^1, ..., G_o^{N_o}\}$ from a sequence of RGBD images, $I_o^t \in \mathbb{R}^{H_{img} \times W_{img} \times 4}, \forall t = \{0, ..., T\}$, where N_o denotes the number of 3D Gaussians, G_o^k represents an individual 3D Gaussian, and H_{img} and W_{img} correspond to the height and width of each RGBD frame. T is the length of the sequence used to generate the original 3D Gaussian splatting representation. As our focus is on representing 3D spaces rather than 3D objects, we use a 3D Gaussian splatting-based SLAM method [12], instead of the vanilla 3D Gaussian splatting [6], which is primarily intended for object-centered 3D reconstruction. As a result of this process, we obtain a set of 3D Gaussians that represent the 3D space, each associated with 3D position $\boldsymbol{p}$, color $\boldsymbol{c}$, opacity $\boldsymbol{o}$, normal vector $\boldsymbol{n}$, and rotation vector $\boldsymbol{q}$.

$$G_o^k = \{\boldsymbol{p}_k, \boldsymbol{c}_k, \boldsymbol{o}_k, \boldsymbol{n}_k, \boldsymbol{q}_k\}, \tag{1}$$

where G_o^k is the k^{th} 3D Gaussian in $\mathbb{M}_o$.

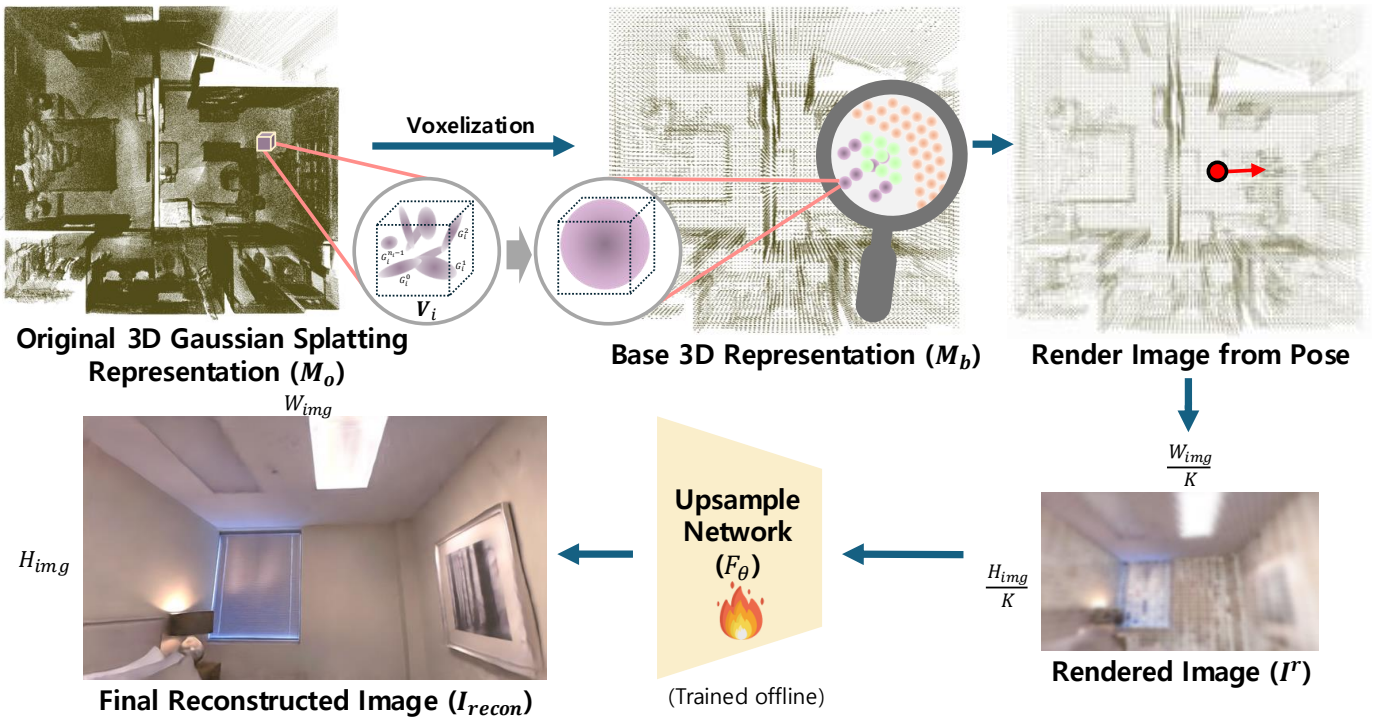

Fig. 2. Based on the full 3D Gaussian splatting representation $\mathbb{M}_o$, we downsample it by voxelization to create base 3D representation $\mathbb{M}_b$. When a specific pose is provided, the proposed method renders a sparse, $1/K$ low-resolution image using $\mathbb{M}_b$ and reconstructs it to its original resolution through an upsampling network.

Next, we optimize the original 3D Gaussians $\mathbb{M}_o$ through voxelization to construct the base 3D Gaussians $\mathbb{M}_b = \{G_b^1, ..., G_b^{N_b}\}$, a sparsified version of $\mathbb{M}_o$, where N_b denotes the number of reduced 3D Gaussians ($N_b < N_o$). During voxelization, we aggregate the color and opacity values of 3D Gaussians within each voxel by averaging, representing the RGB value $\boldsymbol{c}_i$ and opacity $\boldsymbol{o}_i$ of the voxel.

$$\begin{cases} \boldsymbol{c}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \boldsymbol{c}_j, \\ \boldsymbol{o}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \boldsymbol{o}_j, \end{cases} \quad \forall G_i^j \in \boldsymbol{V}_i, \tag{2}$$

where $\boldsymbol{V}_i = \{G_i^0, G_i^1, ..., G_i^{n_i-1}\}$ represents the i^{th} voxel, and n_i is the number of 3D Gaussians within $\boldsymbol{V}_i$. Each 3D position of 3D Gaussian is aggregated to fit within a sphere centered at the center of its corresponding voxel. Therefore, the normal vector $\boldsymbol{n}$ and the rotation vector $\boldsymbol{q}$ are no longer necessary. The voxel size determines the diameter of each 3D Gaussian, and we balance memory efficiency with scene fidelity by inductively selecting an optimal voxel size.

b) Upsample Network: Rendering novel view images from $\mathbb{M}_b$ produces a coarse depiction of the scene, as the base 3D Gaussians lacks detailed 3D Gaussian points necessary for creating high-quality renderings. To address this, we incorporate an upsampling neural network layer F_θ to reconstruct the high-quality images. Unlike existing super-resolution methods, which typically target blurred or low-resolution images, our approach focuses on low-quality rendered images that retain coarse detail. To address this, we generate our own dataset consisting of low-quality rendered images from the voxelized base 3D Gaussians $\mathbb{M}_b$ and their corresponding high-quality ground-truth images, and train the network specifically for our purpose. As illustrated, the low-quality images exhibit sphere-like renderings due to the coarse representation of the base 3D Gaussians. Additionally, these images are rendered at a low resolution of the ground-truth images.

For the upsampling network structure, we employ an effective and lightweight model based on SRResNet [21]. Since the goal of the proposed method is to create a memory-efficient spatial representation, we use an upsampling network with minimal parameter consumption to save memory. The upsampling network is trained using the following loss function:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(I_i^{gt} - F_\theta(I_i^r))^2, \tag{3}$$

where $I^{gt} \in \mathbb{R}^{H_{img} \times W_{img} \times 3}$ and $I^r \in \mathbb{R}^{\frac{H_{img}}{K} \times \frac{W_{img}}{K} \times 3}$ denote the ground truth images and rendered images from $\mathbb{M}_b$, respectively. K denotes the scaling factor applied to the input image of the upsampling network. Unlike the original methods, our training inputs, I^r, incorporate two key characteristics: low resolution and coarse renderings derived from sparse 3D Gaussian splatting representation generated through voxelization.

c) 3DSR: After creating the base representation $\mathbb{M}_b$ and training the upsampling neural network F_θ, we define the voxelized renderable neural 3D spatial representation as $\mathbb{M}_{3DSR} = \{\mathbb{M}_b, \theta\}$, where θ represents the parameters of the F_θ network. Base on the $\mathbb{M}_{3DSR}$, the final reconstructed image I_{recon} from a given 6-DoF pose $\boldsymbol{p}_{query}$ is derived as follows:

$$I_{recon} = F_\theta(F_{render}(\mathbb{M}_b, \boldsymbol{p}_{query})). \tag{4}$$

F_{render} denotes the 3D Gaussian splatting rendering function.

B. Robotic Applications

We demonstrate the practicality of our method in robotics by evaluating it on two core robotics tasks—visual localization and visual navigation—showing that it can be deployed efficiently in real-world applications. These experiments highlight two key strengths: (1) memory efficiency achieved through a compact 3D Gaussian representation, and (2) effective data augmentation enabled by its renderability.

a) Visual Localization: We propose a visual localization framework that effectively leverages the *memory-efficient* and *renderable* properties of 3DSR. The framework comprises two stages: coarse localization through visual feature matching, followed by pose refinement using photometric loss.

First, we extract the floor region from the 3DSR to construct a localization layer $\mathbb{M}_{loc}$, which resembles a 2D bird's-eye-view voxel map. Unlike feature-embedded 3D Gaussian splatting methods [22] which can be memory-intensive due to per-Gaussian feature storage, our voxelized representation improves memory efficiency while preserving localization-related information. For each voxel point in $\mathbb{M}_{loc}$, we capture C images from diverse viewpoints and extract visual features from these observations. The extracted C visual features are appended to each voxel point in $\mathbb{M}_{loc}$. This feature embedding enables our method to estimate the coarse camera pose by matching query image features with the most similar features stored in the $\mathbb{M}_{loc}$.

The result of this initial coarse localization is a discrete pose estimate, which needs further refinement for precise pose determination. To achieve this precision, we integrate a rendering-based fine localization module. This module refines the initial coarse pose by sampling multiple nearby poses and evaluating their accuracy through photometric loss. As proposed in [23], [24], this loss is calculated as $\mathcal{L} = \frac{1}{M}\sum_{i=1}^{M}|I_i^q - F_\theta(I_i^r)|$, where I^q is the query image and M is the number of pixels. The function F_θ renders an image (I^r) from a given pose, allowing us to compare the query image (I^q) to the rendered view. The pose that yields the minimum photometric loss is ultimately selected as the final, precise pose estimate.

b) Visual Navigation: The visual navigation task emphasizes the efficiency in data augmentation of 3DSR. Recent advances in visual navigation have shifted from traditional map-based methods toward general navigation models [25]–[27] that follow data-driven, end-to-end learning paradigms. These methods address image-goal navigation using a behavior-cloning approach: given a target image and a sequence of visual observations, they output an action policy that drives the agent to the goal. Thus, training these models demands substantial quantities of image–action pair data. Since 3DSR can synthesize novel views from arbitrary poses, it readily generates the additional image-action pairs needed for training. To evaluate this capability, we adopt NoMaD [25], a state-of-the-art general navigation model that typically relies on extensive data collections for training. We compare two versions of NoMaD: baseline model and a version fine-tuned with 3DSR-generated data. The baseline model is pre-trained on the GNM [27] and SACSoN [28] datasets. We then fine-tune this model with random images generated from 3DSR constructed on the HM3D dataset [19]. By comparing the baseline and the 3DSR-augmented fine-tuned versions of NoMaD, we demonstrate the effectiveness of 3DSR for enhancing visual navigation performance.

IV. Experiment

A. Experimental Setup

a) Dataset: We use Replica dataset [17], TUM dataset [18], and HM3D dataset [19] for the experiments. The Replica dataset provides synthetic indoor scenes with accurate depth and RGB images. For our experiments, we utilize 2,000 RGB-depth pairs as specified in the experimental setup of [9]. The TUM dataset contains real-world RGB and depth images captured under challenging conditions, including significant noise, motion blur, and information loss in depth data. We evaluate the impact of the proposed method on memory reduction and image quality on real-world scenarios by comparing original dense representation ($\mathbb{M}_o$) and 3DSR. The HM3D dataset is a Habitat simulator-based [29] dataset captured from real-world house environments. It covers larger spaces than the Replica and TUM dataset, making it suitable not only for image quality evaluation but also for vision-based robotics application experiments.

b) Base 3D Gaussians: To create $\mathbb{M}_o$ using 3D Gaussian splatting-based SLAM, we employ GS-ICP-SLAM [12] for its rapid, dense representation and high image quality. Based on $\mathbb{M}_o$, we generate $\mathbb{M}_b$, which serves as the foundational representation for our proposed method. This transformation yields a substantial memory saving, achieving an average reduction of 98.25% compared to the original $\mathbb{M}_o$. However, the proposed method is not limited to a specific approach and can be applied to any type of 3D Gaussian splatting-based representations. Any method capable of generating a point cloud with 3D Gaussians is compatible with 3DSR, making it an efficient solution for memory-efficient 3D Gaussian splatting applications.

c) Upsampling Network: The backbone of the upsampling network is based on the SRResNet [21], which is trained offline. For the Replica and TUM dataset, we use the provided ground truth images for training, which limits the image size to the dimensions of the ground truth images. Specifically, in the Replica dataset, the target image size is 1200×680 and in the TUM dataset, we use 640×480 sized images. For the HM3D dataset, we use rendered images from the full 3D Gaussian splatting representation as ground truth for training, with a training image size of 640×480. We set the scaling factor K for the network to four. Therefore, the training input image sizes for the datasets are 300×170, 160×120, and 160×120, respectively.

d) Evaluation: We validate the effectiveness of the proposed method by evaluating with the neural field-based SLAM methods [8], [9], [11]–[13], which primarily focus on reconstructing 3D spaces. We also evaluate against compact or compressed 3D Gaussian splatting methods [14], [15]. These compact or compressed 3DGS approaches rely on pose estimates obtained by COLMAP [30], as does the original 3DGS [6]. However, COLMAP primarily addresses structure-from-motion in object-centered settings and does not converge on

TABLE I
EVALUATION OF MEMORY CONSUMPTION AND IMAGE RENDERING QUALITY ON REPLICA DATASET.

Method	Metrics	R0	R1	R2	O0	O1	O2	O3	O4	Avg.
3DSR (Proposed)	Mem. (MB) ↓	**20.22**	**19.52**	**19.62**	**19.22**	**18.89**	**19.92**	**21.22**	**20.22**	**19.85**
	PSNR (dB) ↑	32.09	33.21	33.15	37.19	39.42	33.59	33.40	35.44	34.69
	SSIM ↑	0.900	0.904	0.918	0.943	0.947	0.904	0.907	0.931	0.919
	LPIPS ↓	0.223	0.214	0.237	0.211	0.226	0.273	0.235	0.216	0.229
GS-ICP-SLAM [12]	Mem. (MB) ↓	104.38	103.50	103.12	103.40	104.57	103.12	103.69	106.94	104.09
	PSNR (dB) ↑	32.20	**35.36**	34.42	40.31	40.75	33.85	34.08	36.47	35.93
	SSIM ↑	0.940	0.960	0.957	0.978	0.977	0.962	0.953	0.963	0.962
	LPIPS ↓	0.081	**0.067**	0.083	0.045	0.051	**0.069**	0.067	**0.065**	**0.066**
Gaussian Splatting SLAM [11]	Mem. (MB) ↓	42.42	30.25	30.78	34.51	28.78	42.55	48.71	33.08	36.39
	PSNR (dB) ↑	**36.30**	34.02	38.56	**43.06**	**43.32**	37.29	**37.13**	38.28	**38.50**
	SSIM ↑	0.961	0.933	0.969	**0.983**	0.981	0.964	**0.964**	0.965	0.965
	LPIPS ↓	0.072	0.146	0.078	**0.043**	**0.050**	0.089	**0.065**	0.082	0.078
SplaTAM [13]	Mem. (MB) ↓	331.96	434.24	369.23	432.80	365.73	293.00	342.42	344.43	364.23
	PSNR (dB) ↑	32.60	33.55	34.83	38.09	39.02	31.95	29.53	31.55	33.89
	SSIM ↑	**0.975**	**0.969**	**0.982**	0.982	**0.982**	**0.966**	0.949	0.951	**0.970**
	LPIPS ↓	**0.070**	0.097	**0.074**	0.088	0.093	0.098	0.119	0.150	0.099
Compressed 3DGS [15]	Mem. (MB)	384.04	219.00	216.26	93.39	66.23	267.64	179.86	147.71	196.77
	PSNR (dB) ↑	34.98	33.99	**38.67**	42.10	41.92	**37.48**	37.05	**39.92**	38.27
	SSIM ↑	0.949	0.942	0.965	0.978	0.966	0.963	0.961	**0.966**	0.962
	LPIPS ↓	0.117	0.148	0.127	0.085	0.178	0.144	0.122	0.123	0.131
Compact 3DGS [14]	Mem. (MB)	90.52	73.21	60.13	57.60	52.40	67.74	60.91	58.70	65.15
	PSNR (dB) ↑	35.50	34.11	38.19	41.99	41.77	36.99	36.44	39.18	38.03
	SSIM ↑	0.953	0.942	0.964	0.977	0.967	0.961	0.958	0.964	0.961
	LPIPS ↓	0.102	0.146	0.128	0.086	0.169	0.145	0.125	0.124	0.128
NICE-SLAM [9]	Mem. (MB) ↓	94.34	81.43	163.24	135.07	130.29	90.33	105.72	113.50	114.24
	PSNR (dB) ↑	23.17	23.05	23.60	28.12	28.75	20.61	21.42	26.14	24.36
	SSIM ↑	0.708	0.767	0.809	0.872	0.872	0.819	0.803	0.872	0.815
	LPIPS ↓	0.409	0.361	0.338	0.323	0.321	0.367	0.342	0.308	0.345
iMAP [8]	Mem. (MB) ↓	48.82	48.82	48.82	48.82	48.82	48.82	48.82	48.82	48.82
	PSNR (dB) ↑	13.97	18.92	20.37	21.42	26.80	12.09	15.53	19.35	18.55
	SSIM ↑	0.586	0.731	0.773	0.799	0.871	0.712	0.725	0.818	0.752
	LPIPS ↓	0.521	0.393	0.358	0.378	0.301	0.473	0.420	0.361	0.401

TABLE II
EVALUATION OF MEMORY CONSUMPTION AND SPATIAL REPRESENTATION QUALITY ON TUM DATASET.

Method	Metrics	fr1/desk	fr2/xyz	fr3/office	Average
3DSR (Proposed)	Mem. ↓	**18.62**	**18.89**	**25.22**	**20.91**
	PSNR ↑	**25.84**	22.51	**25.79**	**24.71**
	SSIM ↑	**0.788**	0.730	**0.802**	**0.773**
	LPIPS ↓	0.356	0.357	0.328	0.347
GS-ICP SLAM ($\mathbb{M}_o$) [12]	Mem. ↓	35.59	136.75	146.86	106.4
	PSNR ↑	17.69	**23.41**	20.61	20.57
	SSIM ↑	0.706	**0.834**	0.139	0.560
	LPIPS ↓	**0.305**	**0.139**	**0.232**	**0.225**

our evaluation dataset, which covers room-sized spatial data. Therefore, we employ hierarchical localization [31], which is designed for estimating poses in large-scale environments, to recover the input image poses.

For image quality evaluation, we adopt three standard photometric rendering metrics: peak signal-to-noise ratio (PSNR, dB), structural similarity index measure (SSIM) [32], and learned perceptual image patch similarity (LPIPS) [33], along with memory consumption (MB). We measure the memory consumption of the rendering-related representation, including the neural network parameters (θ) for the methods that utilize them. This includes the memory of the `.ply` file and the network parameter file where applicable. For compressed 3DGS, we use the original conversion code provided by the authors to transform `.npy` files into `.ply`.

B. Image Reconstruction

Figure 3 shows the qualitative results of the proposed method on the real-world dataset. The proposed method produces a much clearer background image, whereas GS-ICP-SLAM shows it as a white, blurred patch. Likewise, while GS-ICP-SLAM renders the dice and magazine on the table indistinctly, our method depicts them sharply. We further substantiate these qualitative improvements through quantitative analysis. Table I presents the overall quantitative results of the image reconstruction experiment on Replica dataset and Table II summarizes the results on TUM dataset. The proposed method achieves the lowest memory usage across both real-world (TUM) and synthetic (Replica) datasets while maintaining image rendering quality comparable to existing methods. The results for the TUM dataset show that the proposed method is robust on real-world noisy data. Moreover, Table II shows that the proposed method achieves 120.14% of the image reconstruction quality of the original dense 3D Gaussian splatting representation ($\mathbb{M}_o$) on real-world data

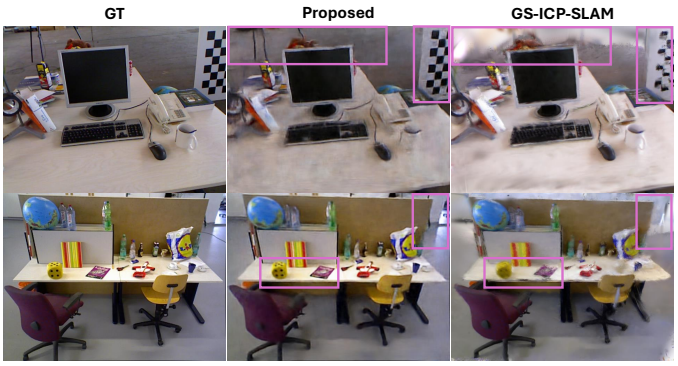

Fig. 3. Qualitative results of the proposed method on the TUM dataset, which contains images captured from the real world.

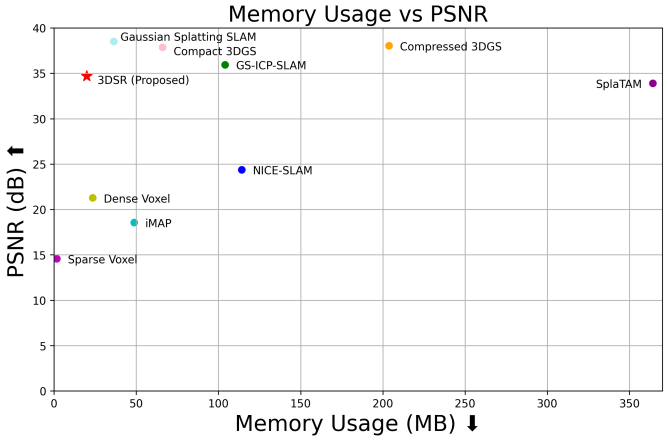

Fig. 4. Memory (MB) - PSNR (dB) plot for experiments on Replica dataset.

TABLE III
ATE AND ARE ON THE PROPOSED SPATIAL REPRESENTATION, VOXELIZED 3D GAUSSIANS ($\mathbb{M}_b$), AND FULL 3D GAUSSIAN SPLATTING REPRESENTATION ($\mathbb{M}_o$).

	3DSR		$\mathbb{M}_b$	$\mathbb{M}_o$
	/w fine loc.	/wo fine loc.		[12]
ATE (m)	0.6796	0.6796	0.6947	**0.6747**
ARE (rad)	**0.2823**	0.4452	0.3464	0.2928

with 19.65% of its memory, demonstrating its practicality for robotic applications.

Figure 4 compares the memory consumption and image reconstruction quality (PSNR) of the evaluated methods on the Replica dataset. The proposed method outperforms NeRF-based methods in both memory consumption and image quality, while achieving comparable image quality to fully detailed 3D Gaussian splatting methods. SplaTAM, despite its high-quality output, is even more memory-intensive, consuming 18 times the memory of 3DSR. Specifically, 3DSR achieves 90.10% of the image quality (PSNR) of the best-performing method (Gaussian Splatting SLAM) while consuming only 54.54% of the memory. Additionally, the proposed method achieves 102.36% of the image quality (PSNR) of SplaTAM while using only 5.45% of the memory, and 96.55% of the image quality of GS-ICP-SLAM while using only 19.07% of the memory.

In addition, we evaluate the inference time of each module on both the Jetson Orin Nano Super and a desktop to demonstrate the practical applicability of our method in robotics. On the Jetson Orin Nano Super platform, the average inference time per image is 0.234 seconds, consisting of 0.008 seconds for rendering from $\mathbb{M}_b$ and 0.226 seconds for the upsampling network. On the desktop, the average inference time per image is 0.031 seconds, with 0.003 seconds for rendering and 0.028 seconds for the upsampling network. The maximum GPU memory allocated during inference is 1.741 GB.

C. Robotic Applications

We evaluate our method on visual localization and navigation tasks using the Habitat–Matterport3D (HM3D) dataset [19] with Habitat simulator [29], a benchmark specifically designed for indoor, vision-based robotic applications. We randomly select 10 environments from HM3D to conduct our experiments.

a) Visual Localization: We evaluate the localization performance of the proposed method using both the original 3D Gaussian splatting representation with full 3D Gaussian points $\mathbb{M}_o$ and the voxelized representation $\mathbb{M}_b$. All three spatial representations share the same $\mathbb{M}_{loc}$ for coarse localization. Therefore, this experiment primarily evaluates the impact of rendering quality on localization accuracy. We randomly select 100 points in each environment for evaluation. The evaluation follows standard localization metrics. For the average translational error (ATE), we compute the average Euclidean distance between the estimated pose (p_{est}) and the ground truth pose (p_{gt}) across the validation set: $ATE = \frac{1}{N_v}\sum_{i=1}^{N_v} ||p_{est} - p_{gt}||_2$. For the average rotational error (ARE), we adopt the rotation averaging metric, which computes the minimum angle ϕ that satisfies the following equation: $2cos(|\phi|) = tr(\mathbf{R}_{gt}^{-1}\mathbf{R}_{est}) - 1$. Base on the derived ϕ, we compute the average value: $ARE = \frac{1}{N-v}\sum_{i=1}^{N_v}\phi$.

The results, presented in Table III, demonstrate that the proposed method achieves performance comparable to or even better than the full 3D Gaussian splatting representation. Furthermore, to evaluate the effectiveness of photometric loss-based fine localization module, we compare results before and after its implementation. The module significantly reduces rotational error by an average of 36.59%, decreasing from 0.4452 rad without fine localization to 0.2823 rad with it. This result highlights that the renderability of the proposed spatial representation is beneficial for solving visual localization tasks.

b) Visual Navigation: In our visual-navigation experiments, we demonstrate the efficiency of 3DSR as a data-augmentation. We assess visual navigation performance with RVN-Bench [34] by contrasting a baseline NoMaD model (pre-trained) with a version fine-tuned using data generated by 3DSR. Assuming a discrete 3-DoF action space $(x, y, \theta_{heading})$ and a fixed sensor height, we use 3DSR to randomly render additional training set (trajectory images) for fine-tuning. Performance is measured using two standard metrics for image-goal navigation: success rate and success

TABLE IV
ABLATION STUDY ANALYZING THE IMPACT OF UPSAMPLING NETWORK CONFIGURATIONS AND VOXEL SIZE VARIATIONS ON MEMORY EFFICIENCY AND IMAGE QUALITY.

	Metric	No Upsample	Patchwise Upsample	Upsample $\times 1$	**Upsample $\times 4$**
Sparse Voxel	Memory (MB) ↓	**1.77**	14.30	14.30	19.85
	PSNR (dB) ↑	14.56	21.55	20.69	**34.69**
	SSIM ↑	0.685	0.780	0.753	**0.919**
	LPIPS ↓	0.507	0.427	0.439	**0.229**
Dense Voxel	Memory (MB) ↓	23.60	38.23	38.23	41.62
	PSNR (dB) ↑	21.27	25.13	24.70	34.00
	SSIM ↑	0.775	0.825	0.823	0.912
	LPIPS ↓	0.432	0.394	0.363	0.241

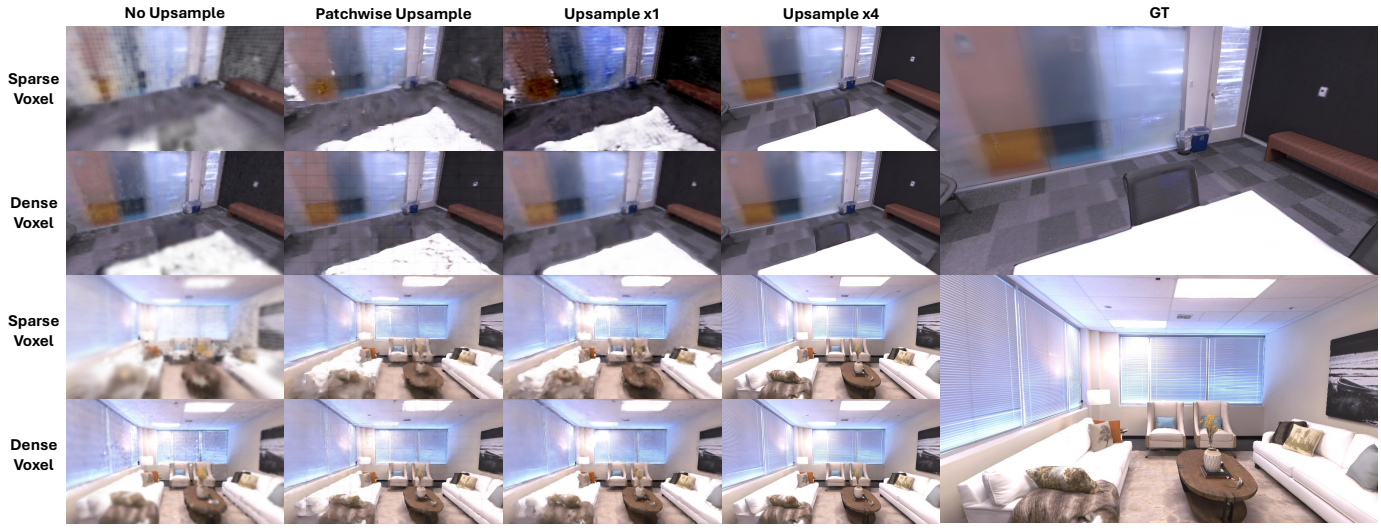

Fig. 5. Qualitative results from the ablation study demonstrate that the upsampling module significantly enhances image quality, with ×4 upsampling producing renderings most similar to the ground truth.

TABLE V
SUCCESS RATE RESULTS FOR VISUAL NAVIGATION TASK.

Augmented Data	NoMaD Pre-trained	**Fine-tuning with 3DSR** 1,000 traj.	5,000 traj.
Success Rate ↑	0.4950	0.5644	**0.6139**
SPL ↑	0.3635	0.4740	**0.5032**

weighted by path length (SPL) [35]. To study the impact of data volume, we progressively increase the amount of 3DSR-augmented trajectory data during fine-tuning. The results are presented in Table V. The fine-tuned model outperforms the pre-trained baseline, demonstrating that the 3DSR-augmented data effectively enhances visual navigation performance. Moreover, even a small amount of augmented data (1,000 trajectories) by 3DSR yields a measurable performance gain over the baseline.

D. Ablation Study

We conduct additional analyses by varying the configurations of the upsampling network (no upsampling module, patchwise upsampling, upsampling with a factor of $\times 1$, and upsampling with a factor of $\times 4$) and by varying the voxel size (sparse and dense). The $\times 4$ upsampling factor was adopted from the original SRResNet paper [21], which we used as the basis for our upsampling module. First, we evaluate the method without the upsampling network, where only voxelization is applied, which is the same as $\mathbb{M}_b$. Second, for patchwise upsampling, the input image is divided into multiple patches, which are processed in batches for training the network. Since most lightweight upsampling networks are designed for small-sized images, such as 96×96, we assume that training an upsampling network with small patches can improve performance. Third, in the upsampling with $\times 1$ configuration, the source and ground truth images used for training the network have the same resolution. For the sparse voxels configuration, we set the voxel size to 10 cm, while for the dense voxel configuration, we set the voxel size to 2.5 cm. We compare these settings with the proposed method, which uses an upsampling factor of $\times 4$.

Figure 5 and Table IV shows the qualitative and quantitative results of the ablation study. Images generated with dense voxels show higher quality compared to those with sparse voxels, and the upsampling module significantly enhances performance. However, dense voxels involve a larger number of 3D Gaussian points, resulting in higher memory consumption compared to sparse voxels. Additionally, the upsampling network with $\times 1$ shows low quality renderings, and patchwise

upsampling results in noisy renderings with visible borders between patches during the assembly process. The proposed method, which employs sparse voxels with $\times 4$ upsampling, achieves an optimal balance between high memory efficiency and excellent image quality.

V. Conclusion

This paper introduces 3DSR, a memory-efficient approach for representing spatial information via 3D Gaussian splatting. By combining voxelization and an upsampling network combined with 3D Gaussian splatting, the proposed method produces high-quality images while significantly reducing memory usage compared to existing methods. However, several limitations remain. These include poor generalization of the upsampling network, a loss of detail due to coarse voxelization, and room for optimization, such as CUDA parallelization. Addressing these limitations in future work could further enhance the efficient contextual understanding of robots.

References

[1] G. Chen, W. Dong, P. Peng, J. Alonso-Mora, and X. Zhu, "Continuous occupancy mapping in dynamic environments using particles," *IEEE Transactions on Robotics*, vol. 40, pp. 64–84, Oct. 2023.

[2] K. Stepanas, J. Williams, E. Hernández, F. Ruetz, and T. Hines, "Ohm: Gpu based occupancy map generation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 078–11 085, Aug. 2022.

[3] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, J. Zhou, and J. Lu, "Surroundocc: Multi-camera 3D occupancy prediction for autonomous driving," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Oct. 2023.

[4] X. Zhao, B. Chen, M. Sun, D. Yang, Y. Wang, X. Zhang, M. Li, D. Kou, X. Wei, and L. Zhang, "Hybridocc: Nerf enhanced transformer-based multi-camera 3D occupancy prediction," *IEEE Robotics and Automation Letters*, vol. 9, no. 9, pp. 7867–7874, Jun. 2024.

[5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "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, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, Jul. 2023.

[7] T. Hua and L. Wang, "Benchmarking implicit neural representation and geometric rendering in real-time rgb-d slam," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2024.

[8] E. Sucar, S. Liu, J. Ortiz, and A. Davison, "iMAP: Implicit mapping and positioning in real-time," in *Proc. of the Int. Conf. on Computer Vision (ICCV)*, Oct. 2021.

[9] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "NICE-SLAM: Neural implicit scalable encoding for slam," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022.

[10] X. Kong, S. Liu, M. Taher, and A. J. Davison, "vMAP: Vectorised object mapping for neural field slam," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023.

[11] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian Splatting SLAM," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2024.

[12] S. Ha, J. Yeon, and H. Yu, "RGBD GS-ICP SLAM," in *Proc. of the European Conf. on Computer Vision (ECCV)*, Oct. 2024.

[13] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "SplaTAM: Splat, track & map 3D Gaussians for dense RGB-D SLAM," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2024.

[14] J. C. Lee, D. Rho, X. Sun, J. H. Ko, and E. Park, "Compact 3D Gaussian representation for radiance field," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2024.

[15] S. Niedermayr, J. Stumpfegger, and R. Westermann, "Compressed 3D Gaussian splatting for accelerated novel view synthesis," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2024.

[16] X. Liu, X. Wu, P. Zhang, S. Wang, Z. Li, and S. Kwong, "Compgs: Efficient 3d scene representation via compressed gaussian splatting," in *Proc. of the ACM Int. Conf. on Multimedia*, Oct. 2024, pp. 2936–2944.

[17] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe, "The Replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, Jun. 2019.

[18] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, Oct. 2012.

[19] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. Chang, M. Savva, Y. Zhao, and D. Batra, "Habitat-matterport 3D dataset (hm3d): 1000 large-scale 3d environments for embodied ai," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, Dec. 2021.

[20] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Proc. of the Robotics: Science and Systems (RSS)*, Jun. 2009.

[21] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[22] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi, "Feature 3dgs: Supercharging 3D Gaussian splatting to enable distilled feature fields," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2024.

[23] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, "Loc-nerf: Monte carlo localization using neural radiance fields," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2023.

[24] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "iNeRF inverting neural radiance fields for pose estimation," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Sep. 2021.

[25] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2024.

[26] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "ViNT: A foundation model for visual navigation," in *Proc. of the Conf. on Robot Learning (CoRL)*, Nov. 2023.

[27] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "Gnm: A general navigation model to drive any robot," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2023.

[28] N. Hirose, D. Shah, A. Sridhar, and S. Levine, "Sacson: Scalable autonomous control for social navigation," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 49–56, Nov. 2023.

[29] A. Szot *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," in *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, Dec. 2021.

[30] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[31] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[33] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[34] J. Lee, J. Heo, G. Lee, H. Jun, J. Oh, and S. Oh, "RVN-Bench: A benchmark for reactive visual navigation," *arxiv preprint arXiv:2603.03953*, 2026.

[35] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, Jul. 2018.