Autonomous Robots (2026) 50:8
https://doi.org/10.1007/s10514-025-10229-0

Robust robotic exploration and mapping using generative occupancy map synthesis

Lorin Achey[1] · Alec Reed[1] · Brendan Crowe[1] · Bradley Hayes[1] · Christoffer Heckman[1]

Received: 11 March 2025 / Revised: 18 September 2025 / Accepted: 30 October 2025 / Published online: 29 December 2025
© The Author(s) 2025

Abstract

We present a novel approach for enhancing robotic exploration by using generative occupancy mapping. We implement SceneSense, a diffusion model designed and trained for predicting 3D occupancy maps given partial observations. Our proposed approach probabilistically fuses these predictions into a running occupancy map in real-time, resulting in significant improvements in map quality and traversability. We deploy SceneSense on a quadruped robot and validate its performance with real-world experiments to demonstrate the effectiveness of the model. In these experiments we show that occupancy maps enhanced with SceneSense predictions better estimate the distribution of our fully observed ground truth data (24.44% FID improvement around the robot and 75.59% improvement at range). We additionally show that integrating SceneSense enhanced maps into our robotic exploration stack as a "drop-in" map improvement, utilizing an existing off-the-shelf planner, results in improvements in robustness and traversability time. Finally, we show results of full exploration evaluations with our proposed system in two dissimilar environments and find that locally enhanced maps provide more consistent exploration results than maps constructed only from direct sensor measurements.

Keywords Intelligent systems · Robotics and automation · Artificial intelligence · Diffusion models · Autonomous navigation · Robots · Intelligent robots

1 Introduction

Efficient and reliable robotic exploration is fundamental to the practical deployment of robotic systems for use in real-world scenarios. In order to intelligently explore, an autonomous agent must have a useful representation of its surroundings and be able to localize within the environment. Robotic agents deployed in search and rescue (Schwarz et al., 2017; Bernard et al., 2011) or disaster response missions (Kawatsuma et al., 2012; Park et al., 2017; Kruijff-Korbayová et al., 2021) rarely have the environment mapped a priori. The robotic agent must operate cautiously as it builds the map from direct sensor observations. This results in slower, less efficient exploration as the agent can only operate over terrain in the sensor field-of-view and large portions of the space may be obscured from that view.

This lack of prior scene information means agents are constantly operating on incomplete information. In addition to limited field-of-view for mapping, there are sensor noise issues, unpredictable dynamic obstacles, obstructions of key information, and visual degradation that an agent may need to reason over to complete a given task. While human agents can filter out noise and fill in gaps in observations by drawing on prior experiences, robots are generally limited to using direct observations without the benefits of prior knowledge or intuition. This results in robotic agents that only take action based on direct observations, a process that further reduces exploration efficiency because of time consuming replanning when informative sensor updates occur.

✉ Christoffer Heckman
christoffer.heckman@colorado.edu

Lorin Achey
lorin.achey@colorado.edu

Alec Reed
alec.reed@colorado.edu

Brendan Crowe
brendan.crowe@colorado.edu

Bradley Hayes
bradley.hayes@colorado.edu

[1] Department of Computer Science, University of Colorado Boulder, 1111 Engineering Dr, Boulder, CO 80309, USA

In related fields, such as self-driving cars, researchers attempt to eliminate the mapping uncertainty using pre-compiled high-definition (HD) maps (Ma et al., 2019). This approach provides a good method of localization at the global level, but there still exists uncertainty and missing information locally. Furthermore, this approach scales poorly, as new HD maps are required for any new operating environment. As discussed previously there are many applications where collating an HD map before navigation is not possible, such as in search and rescue scenarios (Chung et al., 2023) or extra-terrestrial exploration (Verma et al., 2024).

With recent breakthroughs in generative AI, neural networks are able to generate convincing human-like results in various modalities, such as text (Touvron et al., 2023), images (Rombach et al., 2022; Kapelyukh et al., 2023), and audio (Dhariwal et al., 2020). We leverage these models to enable robots to make common-sense inference of geometry that is occluded from view. Our model enables a robotic agent to infer geometry through occupancy prediction beyond an obstruction so that the agent can plan a trajectory beyond the field-of-view similar to the way a human would navigate occlusions using intuition.

SceneSense was first introduced in Reed et al. (2024), where a diffusion-based model was used to generate voxelized predictions of occupancy. Subsequent work improved the efficiency of SceneSense, notably achieving faster training and inference times (Reed et al., 2025). These works established that diffusion models can be used for occupancy prediction but left open questions regarding real-time deployment and map level reasoning.

In this work, we extend the SceneSense framework in three ways. First, we demonstrate real-time occupancy generation on a quadruped platform using a diffusion network. Second, we integrate a probabilistic update rule that fuses diffusion based predictions with an evolving observed map, allowing predictions to accumulate and refine over time. Third, we show that SceneSense can generate realistic occupancy predictions across the full map rather than being restricted to regions local to the robot. This broader predictive capability leads to enhanced global maps which can simplify exploration in real world deployments.

We validate our approach through field experiments with a quadruped platform. Specifically, we (i) compare robot-centric and frontier-centric exploration performance, (ii) evaluate one-shot versus probabilistic map merging, and (iii) demonstrate successful navigation in challenging scenarios that historically degrade robotic perception (e.g. transparent structures, narrow hallways). Finally, we provide detailed reporting of system specifications, compute requirements, training times, experimental design, and system limitations. Code for reproducing our results will be made available at [URL-added-post-publication].

2 Related works

2.1 Scene exploration

An autonomous agent must have a useful representation of its surroundings and be able to localize within the environment to efficiently and intelligently explore. A proven strategy for mapping complex environments for use with autonomous systems is 3D volumetric mapping (Ahmad et al., 2021a, b). OctoMap (Hornung et al., 2013) is a popular framework for 3D mapping due to its memory efficiency and wide adoption in the robotics community (Dang et al., 2020). In the DARPA Subterranean (SubT) Challenge (Chung et al., 2023), Team MARBLE (Biggie et al., 2023) utilized OctoMap on a Boston Dynamics Spot robot and finished 3rd overall.

Other works (Yamauchi, 1997; Yamauchi et al., 1998; Makarenko et al., 2002; González-Banos & Latombe, 2002) have shown considerable success using frontier-based exploration strategies. In Freda and Oriolo (2005), a frontier-based approach is utilized to demonstrate the efficacy of biasing towards unseen areas of a map for more effective mobile robot exploration. Further efficiency improvements to the frontier based exploration strategy are introduced in Keidar and Kaminka (2014). In González-Banos and Latombe (2002), a frontier based method is proposed for safe and efficient map-building by defining a “safe region" around the robot, allowing it to explore and gather new information while ensuring obstacle-free navigation, although it is limited to mapping at a fixed height, making it unsuitable for more complex environments.

In Tranzatto et al. (2022), a volumetric map is incrementally generated from depth sensors and a graph-based planner (Dang et al., 2020) reasons over that map to determine if the terrain is traversable for each vertex of a four-legged robot. This graph traversal requires considerable computation time for determining the status of voxels beneath and around the robot platform. Additionally, this method does not address the issues with path planning if there are obstacles or geometric occlusions in the sensor field of view.

Despite the improvements in mapping and frontier-based exploration strategies, Team MARBLE (Biggie et al., 2023) reported that there are still challenges with efficient exploration. During the SubT challenge, exploring robots frequently wasted valuable time attempting to explore in previously seen areas rather than extending their trajectories beyond their current frontiers (Biggie et al., 2023). Influxes of new information from onboard sensors that came when turning

corners or exploring new rooms caused the robots to pause for extended periods of time for re-planning, slowing the pace of exploration considerably (Biggie et al., 2023). The challenge outcomes demonstrated the utility of 3D volumetric mapping using frontier-based exploration for discovery, but also elucidated the need for increased efficiency in the planning and exploration process for time critical missions. Our novel approach improves the efficiency of exploration by reducing ambiguity that leads to long re-planning delays for the robots.

2.2 Occupancy prediction

Wang et al. (2021) attempts to address challenges of autonomous navigation in environments with occlusions by utilizing a deep learning-based approach to predict the occupancy distribution of portions of the environment. Their method, the Occupancy Prediction Network (OPNet), predicts obstacle distributions, utilizing a self-supervised learning technique that generates training data through simulated navigation trajectories with simulated sensor noise. This approach relies on removing portions of the data from the Matterport3D (Chang et al., 2017) dataset to train the neural network which biases the model to predict occupancy only for the types of occlusions removed from the original dataset. This approach does not scale well to large unseen sections of an environment, is limited to static updates, and does not generalize due to its reliance on a specially curated dataset.

More recently, Huang et al. (2024) introduced SelfOcc, a self-supervised method for 3D occupancy prediction using video sequences. SelfOcc converts 2D images into 3D representations with deformable attention layers and uses signed distance fields (SDF) for regularization and occupancy boundary determination. Although SelfOcc performs well with surrounding cameras, it cannot predict occupancy beyond the camera view due to the limitations of SDF in handling occluded geometry.

2.3 Scene synthesis

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020), are a popular and promising technique which have demonstrated impressive results in image (Rombach et al., 2022), video (Harvey et al., 2022), and natural language (Huang et al., 2022). Based on these successes, diffusion models are being extended to 3D scene and shape generation. Recent work (Luo & Hu, 2021) demonstrates the use of diffusion models for 3D point cloud generation for simple shapes and objects (e.g. tables, chairs). Kim et al. (2023) shows successful 3D shape generation from 2D content such as images, and Vahdat et al. (2022) demonstrates similar 3D shape generation but using point cloud datasets rather than images. In LegoNet (Yu et al., 2022), diffusion models are used to rearrange objects in a 3D scene. In DiffuScene (Tang et al., 2023), a denoising diffusion model is used with text conditioning to generate 3D indoor scenes from sets of unordered object attributes. Unlike these previous works which primarily focus on generating simple shapes, rearranging objects, or creating indoor scenes, our approach leverages diffusion models to fuse generated terrain with the local robot field of view, thereby bridging the gap between 3D scene generation and practical robotics applications.

2.4 Generative AI in robotics

The body of work devoted to applications of generative AI in robotics is steadily growing. Yuan et al. (2023) demonstrates that hierarchical generative modeling, inspired by human motor control, can enable autonomous robots to effectively perform complex tasks with robust performance even in challenging conditions. DALL-E-Bot (Kapelyukh et al., 2023), uses web-scale diffusion models for generating an image from a text prompt which the robot utilizes to rearrange real objects in accordance with the image. Diffusion models were also used in (Carvalho et al., 2023) to improve robot motion planning by learning priors on trajectory distributions from previously successful plans, a methodology which shows good generalization capabilities in environments with previously unseen obstacles. More recently, diffusion models were shown to successfully generate terrain predictions behind occluded geometries in indoor environments using a single RGB sensor mounted on a mobile robot platform (Reed et al., 2024). We build on this work, extending the practical applications of diffusion models by demonstrating a novel method for occupancy prediction which shows significant efficiency gains during exploration.

3 Problem statement and preliminaries

The goal of this work is to create and evaluate a method for dense occupancy prediction that demonstrably enhances a robotic platform's ability to navigate, as measured by reducing the time required to fully explore its environment. The ability to generate occupancy predictions beyond occlusions and at range from a mobile platform introduces the subsequent challenge of how to identify and select frontier points for exploration in these generated occupancy spaces. We present an approach to frontier identification and selection for exploration as part of this work.

3.1 Autonomous robotic exploration

Let $\mathbb{M}$ denote the current occupancy map, constructed from onboard sensor measurements $\mathbb{S}$ and odometry $\mathbb{O}$. The map consists of voxels $m \in \mathbb{M}$, where each voxel is categorized as free ($m \in \mathbb{M}_{free}$), occupied ($m \in \mathbb{M}_{occupied}$), or unknown ($m \in \mathbb{M}_{unknown}$). During robotic exploration we seek to recover the complete map $\mathbb{M}_{comp}$ of a target region $\mathbb{M}_{target}$ from successive measurements from the onboard sensors. While accurate completion and alignment of $\mathbb{M}_{comp}$ with $\mathbb{M}_{target}$ is the primary goal, secondary goals include minimizing exploration time and maximizing volumetric gain per second, provided safety is not compromised.

3.2 Frontier identification and evaluation

We identify and evaluate frontiers of interest in $\mathbb{M}$ that can enhance the robot's understanding of the environment during exploration. In general, "interesting" frontiers will maximize the number of unknown voxels available for occupancy prediction while considering common exploration metrics such as directionality, distance from target, and reachability (Biggie et al., 2023; Dang et al., 2020). These frontiers guide the robot's exploration.

3.3 Dense occupancy prediction

Dense occupancy prediction estimates the probability of occupancy for each voxel $m \in \mathbb{M}$, producing a continuous value in [0, 1] where 0 indicates free and 1 indicates occupied. These predictions are used to hypothesize structure beyond the currently observed map generated by the robots sensor measurements.

3.4 Forward diffusion

Let x_0 denote an occupancy subgrid ($x \subset \mathbb{M}$), sampled from the data distribution $q(x_0)$. By sampling from the data distribution $x_0 \sim q(x_0)$, the forward diffusion process is defined as a Markov chain of variables $x_1, ..., x_T$ that iteratively adds Gaussian noise to the sample. A diffusion step at time *t* in this chain is defined as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1},\ \beta_t I), \tag{1}$$

where t is the time step $t \in [1, T]$, β_t is the variance schedule $0 \leq \beta_t \leq 1$ and *I* is the identity matrix. The joint distribution of the full diffusion process is then the product of the diffusion step defined in Eq. (1):

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}). \tag{2}$$

Conveniently, we can apply the reparameterization trick to directly sample x_t given x_0 using the conditional distribution:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)\mathcal{I}), \tag{3}$$

where $x_t = \sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon$ where $\alpha_t := 1 - \beta_t$, $\overline{\alpha}_t := \prod_{r=1}^{t} \alpha_s$, and ϵ is the noise used to corrupt x_t.

3.5 Reverse diffusion

Reverse diffusion defines a generative process that inverts the forward noising procedure. Specifically, it is modeled as a Markov chain of learned Gaussian transitions, parameterized by a neural network with learnable parameters θ:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{4}$$

where $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are the predicted mean and covariance of the Gaussian distribution over x_{t-1}. The parameters θ correspond to the SceneSense diffusion model, trained to recover occupancy maps from noisy inputs.

Given the initial state of a noisy occupancy map from a standard multivariate Gaussian distribution $x_t \sim \mathcal{N}(0, I)$, the reverse diffusion process iteratively predicts x_{t-1} at each time step *t* until reaching the final state x_0 which is the goal occupancy map. Similar to the Markov chain defined forward diffusion process the joint distribution of the reverse diffusion process is simply the product of the applied learned Gaussian transitions $p_\theta(x_{t-1}|x_t)$:

$$p_\theta(x_{0:T}) := p_\theta(X_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t). \tag{5}$$

4 SceneSense design and implementation

SceneSense (Reed et al., 2024, 2025) is a diffusion model designed to generate 3D occupancy predictions given an observed occupancy map. Notably, SceneSense only generates occupancy predictions in space that has not been observed, and therefore maintains the fidelity of the observed map. In this section we will discuss the design of the SceneSense network as an unconditional diffusion model as well as our method for training. Then we will discuss our inference time control method of occupancy inpainting.

4.1 Network design

4.1.1 Denoising network

We design our denoising network as an unconditional diffusion model. Specifically the denoising network is a U-net constructed from the huggingface diffusers library of blocks (Platen et al., 2022) and consists of Resnet (He et al., 2015) downsampling/upsampling blocks. Notably, as discussed in Reed et al. (2024, 2025) the addition of conditioning for guidance, as often seen in text-to-image diffusion models (Rombach et al., 2022), does not generate more accurate predictions and increases inference times in this particular application. These ablations are presented in Appendix A and B.

4.1.2 Occupancy mapping

Occupancy mapping allows platforms to build a running map of areas that have been measured to contain matter using onboard sensors like lidar or RGB-D cameras. For our framework we use the popular occupancy mapping framework OctoMap (Hornung et al., 2013) to generate an occupancy map as the platform explores the environment. Importantly, OctoMap provides a probability of occupancy $o \in [0, 1]$ for every voxel in the map that has been observed using pose ray casting. This means that as we explore we will maintain not only a map of occupied areas M_o, but also a map of areas that have been measured to not contain any data M_u. These maps will later be used to inform SceneSense where occupancy predictions should be made.

4.2 Training

During training, we generate a noisy local occupancy map x_t where $t \in [1, T]$ from a ground truth local occupancy map x. We train the diffusion model f_θ to predict the noise applied to x_t.

4.2.1 Data augmentation

In previous SceneSense implementations (Reed et al., 2024, 2025), the primary area for prediction was centered around the robot. In this paradigm, the SceneSense training data was collected as local occupancy data collected around the robot. However, this training approach struggles to capture the variety of poses possible when doing occupancy predictions at the frontier. The graph-based planner, where poses are selected, can be much closer to structures like walls, and at a wider variety of rotation poses. To account for this, we augment the training data to more closely approximate that of the graph based poses. For each pose we generate 10 noisy pose samples where noise is added from a uniform distribution of $[-1, 1]$ meters in the x,y direction and $[0, 2\pi]$ in the rotational plane. This modification increases the training set by $10x$ but yields much more consistent results at inference time when performing occupancy prediction around frontier nodes in the graph.

4.2.2 Occupancy corruption

To corrupt each ground truth local occupancy map x to train the network we add Gaussian noise to x to generate x_t. This corruption process is defined in Eq. (3) where the intensity of the noise is controlled by α_t which is configured by a linear noise scheduler (Ho et al., 2020).

4.2.3 Loss function

The network f_θ is trained using the calculated l_2 loss between the denoised x_t prediction and the associated ground truth data x. l_2 loss is a popular diffusion loss function, however other loss functions such as cross-entropy loss or mean squared error can be applied and have had some success in similar diffusion frameworks (Nichol et al., 2022; Rombach et al., 2022; Chen et al., 2022; Ji et al., 2023).

4.3 Inference

4.3.1 Sampling process

The trained noise prediction network f_θ takes isotropic Gaussian noise $\mathcal{N}(0, \mathcal{I})$ as the starting point x_T to begin the reverse diffusion process. The noise is iteratively removed by using f_θ to compute x_{t-1}.

4.3.2 Occupancy inpainting

Our method of occupancy inpainting is provided in Algorithm 1 which ensures observed space is never overwritten with SceneSense predictions. Inspired by image inpainting methods seen in image diffusion (Rombach et al., 2022) and guided image synthesis methods (Meng et al., 2022), occupancy inpainting continuously applies the known occupancy information to the diffusion target during inference. To perform occupancy inpainting we select our target region x as a submap of the full occupancy map. From x we build an occupied map M_o and an unoccupied map M_u composed of observed occupied voxels and observed unoccupied voxels, respectively. From M_u and M_o we generate noisy representations commensurate with the current diffusion step t using Eq. (3), M_{un} and M_{on}, respectively. Finally the data from M_{un} and M_{on} replaces the data in the diffusion target at the associated coordinates.

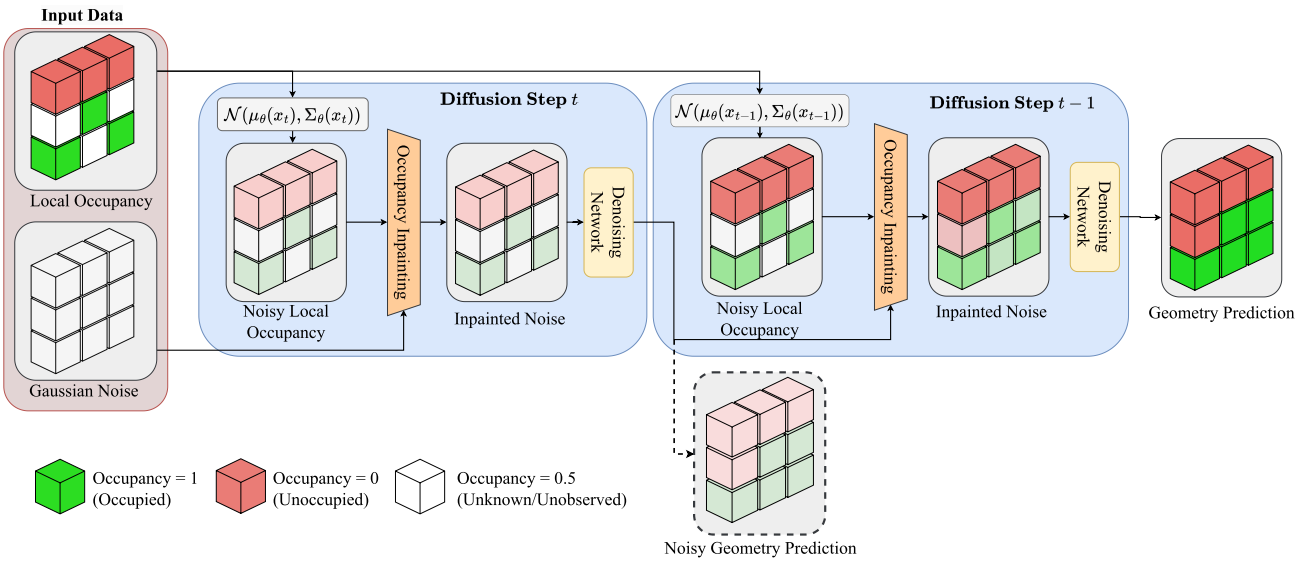


Fig. 1 The reverse diffusion process takes the local occupancy information and the Gaussian noise of the area to be diffused over. Noise commensurate with the current diffusion step is added to the local occupancy information, which includes occupied (green) and observed unoccupied (red) data. The result is inpainted into the noisy local occupancy prediction. The inpainted noise data is provided to the denoising network which generates a new noisy geometry prediction at $t-1$. This process is repeated as the starting noise x_T is iteratively denoised to x_0 which is the final geometry prediction from the framework. This process is further detailed in Algorithm 1 (Color figure online)

This process is repeated for each inference step and can be seen in Fig. 1. This method ensures the diffusion model does not predict or modify geometry in space that has already been observed. This inpainting method is crucial for increasing the fidelity of the scene predictions.

4.3.3 Multiple prediction

Diffusion is a noisy process that can generate different results given the same context. In image generation this is a desirable characteristic as the framework can generate different results given the same prompt, increasing the diversity of the generated data. As shown in Figs. 4 and 9, SceneSense has the same behavior as these networks and can generate different reasonable predictions based on the same input information. Further there is no compute time increase (assuming enough compute is available) as multiple predictions can be done in parallel. We present our method for merging these various occupancy maps in Sect. 6.

5 Robotic system design

Our robotic system is a quadruped (Spot) as shown in Fig. 3 combined with an off-board high performance computer to handle computationally expensive requests. Much of the system design is inherited from the University of Colorado at Boulder MARBLE team that competed in the DARPA Subterranean challenge (Biggie et al., 2023; Chung et al., 2023). For convenience, we provide relevant design decisions in this section and a block diagram of the system in Fig. 2.

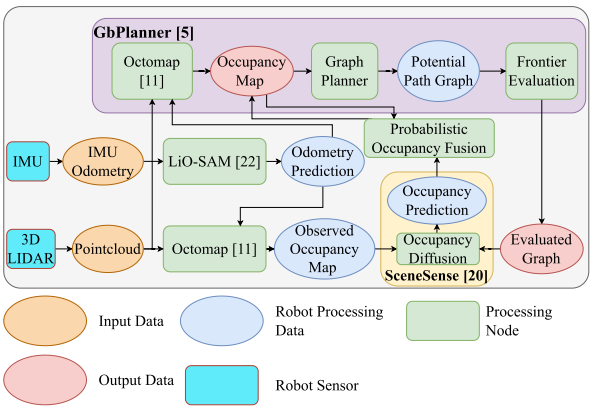


Fig. 2 Block diagram showing the system design for onboard SceneSense occupancy prediction. The system is comprised of an IMU and LIDAR sensor to generate odometry and occupancy maps. Once the occupancy map is built, a graph is constructed to evaluate frontier points for occupancy prediction. Local occupancy is then subselected around these points and sent to the SceneSense framework that provides occupancy predictions. These predictions are then merged with the running occupancy map using the probabilistic update rule

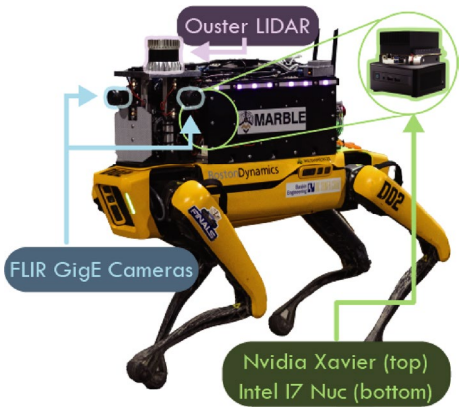


Fig. 3 Spot robotic platform with onboard compute and sensors. Sensor suite consists of a 64 Beam OS1 lidar as well a 3 FLIR GigE Cameras

Require: Unconditional diffusion model $\epsilon_\theta(x_t)$, x_{obs}, m_{obs}

$x_T \sim \mathcal{N}(0, I)$

for $t = T, \ldots, 1$ **do**

In-painting step:

$x_{obs}^{t-1} \sim \mathcal{N}(\sqrt{\alpha_{t-1}} x_{obs}, (1 - \alpha_{t-1})I)$

$x_{t-1}^{m_{obs}} := x_{obs}^{t-1}$

Standard reverse diffusion step:

$\mu_\theta(x_t) := \sqrt{\frac{1}{\alpha_t}} \left(x_t - \frac{\sqrt{\beta_t}}{\sqrt{1-\alpha_t}} \epsilon_\theta(x_t) \right)$

$x_{t-1} \sim \mathcal{N}(\mu_\theta(x_t), \tilde{\beta}_t I)$

end for

return x_0

Algorithm 1 Occupancy Inpainting

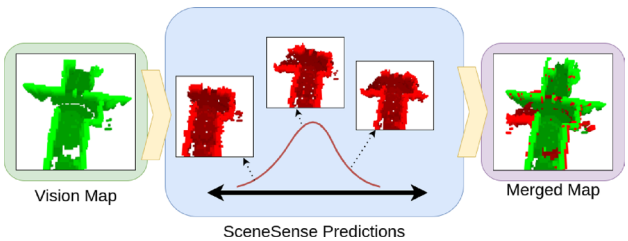


Fig. 4 Multi-Prediction Occupancy Merging: SceneSense predicts various occupancy maps based on equivalent input data that form a distribution. This distribution forms a curve where more likely predictions occur more often, and less likely predictions occur infrequently. These predictions are fused into the merged map using Eq. (7). The resulting merged map naturally filters out the unlikely voxel predictions, forming an extended occupancy map

5.1 Sensor suite

The equipped sensor suite was designed with the purpose of providing 3D point cloud information and sufficient data for accurate localization. The primary sensor in the Spot sensor suite is the Ouster OS1-64 LIDAR which provides 3D point clouds for mapping and localization. In addition, a LORD Microstrain 3DM-GX5-15 IMU is used to measure the linear and angular acceleration of the Ouster, for use in the localization system.

5.2 Localization

Localization is required for Spot to perform volumetric mapping. We have implemented the popular LIDAR-based localization method LIO-SAM (Shan et al., 2020) to provide localization at runtime. To improve localization performance both the IMU and lidar sensor were fastened to a 6061 aluminum sensor plate. This allowed for a high-precision relative transform between the sensors, reducing the necessity of extrinsic calibration. Furthermore, LIO-SAM requires aligned sensor timestamps and sensor publish rates to be constant. The LORD IMU however allows for large fluctuation in publish rate due to the USB transmission delays. To reduce the sensitivity the IMU timestamps are adjusted when messages are not received within 15% of the nominal rate. Additionally, the lidar sensor uses Precision Time Protocol (PTP) to synchronize with the onboard computer. These modifications allowed for consistent localization using LIO-SAM.

5.3 Frontier identification and evaluation

The first work in this domain (Reed et al., 2024) focused on occupancy prediction around the robot. Later work (Reed et al., 2025) updates the occupancy prediction framework to predict occupancy at any point in the map, demonstrating enhanced utility for predictions at range. We leverage this enhancement, identifying interesting areas for prediction using the graph-based exploration planner GBPlanner (Dang et al., 2020). GBPlanner builds a graph where nodes are potential exploration points and edges are feasible paths to navigate from node to node. A ray casting algorithm is run at each node in the graph to quantify the number of observed, free, and unknown voxels in that node's field of view. After finding the shortest path to each node the *Exploration Gain* can be calculated for each node in the graph as follows:

$$\begin{aligned} \text{ExplorationGain}(\sigma_i) &= e^{-\gamma_{\mathcal{S}} \mathcal{S}(\sigma_i, \sigma_{exp})} \\ &\cdot \sum_{j=1}^{m_i} \text{VolumetricGain}(v_j^i) e^{-\gamma_{\mathcal{D}} \mathcal{D}(v_1^i, v_j^i)}, \end{aligned} \tag{6}$$

where $\mathcal{S}(\sigma_i, \sigma_{exp})$, $\mathcal{D}(v_1^i, v_j^i)$ are weight functions with tunable factors $\gamma_{\mathcal{S}}, \gamma_{\mathcal{D}} > 0$ respectively. Furthermore $\mathcal{D}(v_1^i, v_j^i)$ is the cumulative Euclidean distance from a vertex v_j^i to the root v_1^i along a path γ_i.

We use exploration gain to rank nodes where occupancy prediction should run. Given a minimum node spacing d_m

and a maximum number of frontier prediction nodes n_{max}, SceneSense generates occupancy predictions at range, centered around the identified frontiers.

5.4 Mapping

We use the probabilistic volumetric mapping method OctoMap (Hornung et al., 2013) as the system's mapping framework. OctoMap was adopted for its log-odds update method for predicting occupied and unoccupied cells. This approach allows for computationally efficient fusion of observed occupancy and predicted occupancy maps. Further discussion on map fusion is provided in Sect. 6.

6 Probabilistic map merging

In previous work, predicted occupancy was merged into the running map in a "fire and forget" fashion (Reed et al., 2024). Each prediction was incorporated by setting its occupied cells to 1, but when a new prediction was generated, the previous one was discarded and replaced. While simple, this approach prevents the system from maintaining a coherent and continuous estimate of the environment. A single out of distribution prediction can temporarily warp the map, causing the planner to divert the robot along a spurious path. Once the next prediction arrives and overwrites the first, the planner abruptly redirects, potentially producing oscillatory shifts in the planned trajectory. To address these issues, we modify the probabilistic occupancy update formula presented for the OctoMap mapping framework (Hornung et al., 2013).

We define the probability that a voxel m is occupied given an occupancy prediction d_t or sensor reading z_t as $P(m|d_t)$ or $P(m|z_t)$, respectively. The set of sensor estimates $z_{1:t}$ and diffusion estimates $d_{1:t}$ populate the joint set $\{z_{1:t}, d_{1:t}\}$ which we denote as $j_{1:t}$. As discussed in Reed et al. (2024), SceneSense only operates on voxels m that are not contained in the observed set $\mathcal{O}$, where $z_{t:t-1} = \varnothing$, and therefore $P(m|j_{1:t})$ will never require an update given $P(m|d_t)$ and $P(m|z_t)$ at the same time. As such we generate the piece-wise update rule for merging diffusion into the running occupancy map.

$$P(m \mid j_{1:t}) = \begin{cases} \left[1 + \frac{1-P(m|d_t)}{P(m|d_t)} \frac{1-P(m|j_{1:t-1})}{P(m|j_{1:t-1})} \frac{P(m)}{1-P(m)}\right]^{-1} & \text{if } m \notin \mathcal{O} \\ \left[1 + \frac{1-P(m|z_t)}{P(m|z_t)} \frac{1-P(m|j_{1:t-1})}{P(m|j_{1:t-1})} \frac{P(m)}{1-P(m)}\right]^{-1} & \text{if } m \in \mathcal{O}. \end{cases} \tag{7}$$

In this framework $P(m|z_t)$ and $P(m|d_t)$ can be configured to different values prior to runtime. Generally, $P(m|d_t)$ given a predicted occupied cell is set lower than $P(m|z_t)$ given a sensed occupied cell, as we trust the sensor more than our generative model. By using this probabilistic approach to map merging, the final merged map benefits from prediction persistence as the system explores as well as increased map fidelity due to multi-prediction occupancy refinement.

7 Occupancy prediction experiments and evaluations

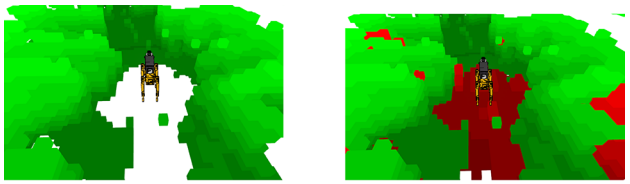
In this section, we provide results and evaluations of the modified SceneSense occupancy prediction framework onboard a real-world system. In particular we evaluate if the SceneSense enhanced occupancy maps better represent a set of fully observed ground truth maps.

Training and Implementation To train SceneSense, we collected real-world occupancy maps from various indoor environments across different buildings. We gathered approximately 1 h worth of occupancy data, resulting in 11, 296 unique poses with associated complete local occupancy maps. Any areas that were used to train the model are omitted from the results presented here.

We implement the same denoising network structure presented in Reed et al. (2024). It is a U-net constructed from the HuggingFace Diffusers library of blocks (Platen et al., 2022) and consists of Resnet (He et al., 2015) downsampling/upsampling blocks. The diffusion model is trained using randomly shuffled pairs of ground truth local occupancy maps x. We use Chameleon cloud computing resources (Keahey et al., 2020) to train our model on one A100 with a batch size of 32 for 250 epochs or 88, 208 training steps. We use a cosine learning rate scheduler with a 500 step warm up from 10^{-6} to 10^{-4}. The noise scheduler for diffusion is set to 1000 noise steps.

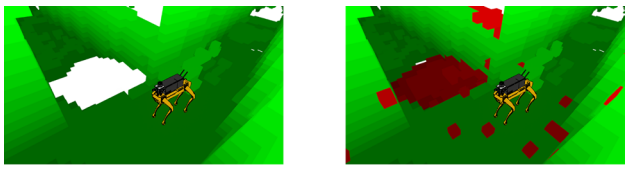
At inference time we evaluate our dataset using an RTX 4070 TI Super GPU for acceleration. The number of diffusion steps is configured to 30 steps.

Evaluation Metrics We evaluate predicted occupancy using a classical computer vision metric, Intersection over Union (IoU) (Everingham et al., 2010), and newer generative modeling metrics, Fréchet Inception Distance (FID) (Heusel et al., 2017) and Kernel Inception Distance (KID) (Bińkowski et al., 2018a). IoU provides a direct measure of overlap between predicted and ground truth occupied voxels. To assess the realism and diversity of generated occupancy predictions, we report FID and KID. FID measures the Fréchet distance between Gaussian statistics of features extracted from predicted and real samples, while KID uses a polynomial kernel Maximum Mean Discrepancy, rather than assuming Gaussian distributions, to provide an unbiased estimate. Both metrics have become standard



(a) (b)

Fig. 5 At startup, the robot cannot observe the ground directly under it due to the mounting location of the lidar. **b** SceneSense generates occupancy predictions (Red Voxels) that fill in the hole under the robot as well as some of the vertical occluded geometry. With the additional predictions, the robot autonomously generates a traversable path and begins exploring without the need for manual intervention via teleoperation (Color figure online)



(a) (b)

Fig. 6 **a** During traversal, the robot fails to observe the ground directly to its right as a result of the mounting location of the lidar and is unable to plan down the hallway for exploration. **b** SceneSense generates occupancy predictions (Red Voxels) that fill in the hole to the right of the robot as well as some of the vertically occluded geometry. With the additional predictions, a plan can now be generated, allowing the robot to continue navigation down the hallway (Color figure online)

in generative modeling, and we compute them using the `clean-fid` library (Parmar et al., 2022).

It is worth noting that generating good metrics to evaluate generative frameworks remains a difficult task (Naeem et al., 2020). FID and KID have become standard metrics for many generative methods due to their ability to score accuracy of predicted results and diversity or coverage of the results when compared to a set of ground truth data. While these metrics are fairly new to robotics, which traditionally evaluates occupancy data with metrics like accuracy, precision and IoU, we report FID and KID, in addition to IoU, because they have been shown to be an effective measure for evaluating predicted scenes (Reed et al., 2024; Tang et al., 2023).

7.1 SceneSense generative occupancy evaluation

For the following experiments we evaluate the occupancy generation capabilities of SceneSense onboard a real-world robot in two unique test environments. In particular, we examine the fidelity of predictions around the robot with predictions at the frontiers of the map, ablating the map update methods and the running sensor only map.

Experimental Setup SceneSense predictions are evaluated in two environments. Environment 1 is a long hallway with cutouts for classrooms and 1 right turn. Select frames shown in Figs. 5 and 6 are from Environment 1. Environment 2 consists of similar carpeted area with 4 hallways and 4 turns, forming a square shape. We evaluate the occupancy prediction framework using the following test configurations.

Table 1 Comparison of Running Occupancy (BL) vs. SceneSense (SS). Evaluations of each method are provided as robot-centric (RC) and frontier-centric (FC) generations

Method	Env. 1		Env. 2	
	FID ↓	KID×1000 ↓	FID ↓	KID×1000 ↓
BL-RC	36.0	16.4	30.3	16.3
SS-RC-OSMM	**26.3**	**7.7**	**20.8**	10.1
SS-RC-PMM	29.2	10.4	21.0	**9.1**
BL-FC	116.9	81.6	150.6	118.8
SS-FC-OSMM	104.2	66.3	133.4	104.4
SS-FC-PMM	**30.1**	**10.3**	**34.5**	**9.0**

Bold indicates best performance within RC or FC generations

1. *Baseline or SceneSense* A comparison between Octomap sensor-only local occupancy (BL) with the SceneSense occupancy prediction included (SS).
2. *Robot-centric or Frontier-centric* Robot-centric diffusion (RC) predicts occupancy at a radius of 3.3m about the robot while frontier-centric diffusion (FC) predicts occupancy at a radius of 3.3m at an identified location in the map, which has a maximum range of 7m from the robot.
3. *One Shot Map Merging or Probabilistic Map Merging* One shot map merging (OSMM) simply takes the current local occupancy map and a SceneSense occupancy prediction and populates the predicted occupancy information in the running map using the “fire and forget" approach described in Sect. 6. Probabilistic map merging (PMM) keeps a running local merged occupancy map that uses update Eq. (7) to update the occupancy map for every new occupancy prediction. In practice, each pose will receive 3–5 SceneSense predictions to merge into the running map.

Generative Occupancy Results Discussion. The results in Table 1 show that RC predictions are quite similar between OSMM and PMM approaches, reducing the FID of the environments by an average of 28.5% and 25% for OSMM and PMM, respectively. These results are similar to the simulation-based results presented in Reed et al. (2024). However, the FC results show that PMM achieves a much greater improvement than OSMM. This improvement is reflected in the average FID, where a lower value indicates better performance. PMM reduces the average FID by 75%, whereas OSMM achieves only an 11% reduction.

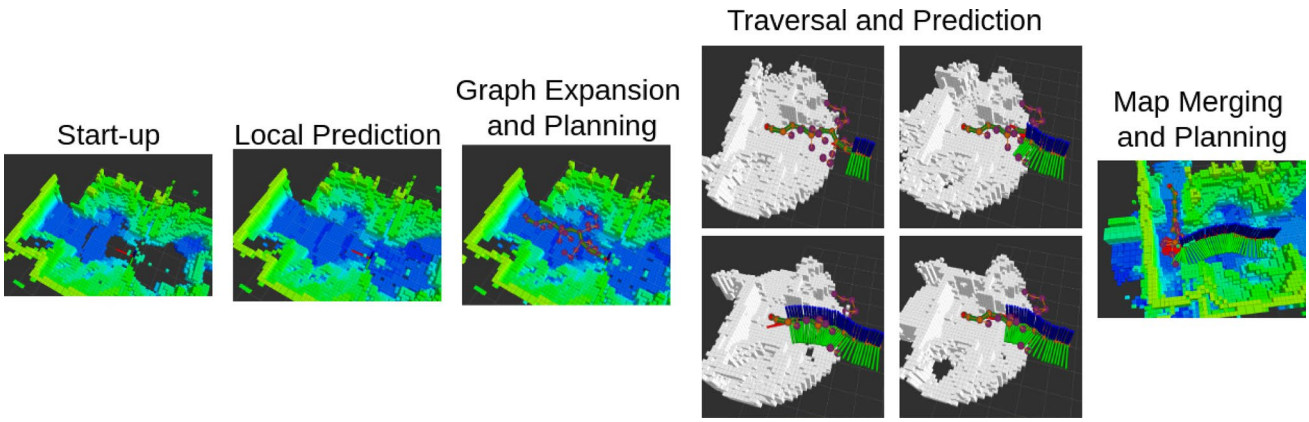


Fig. 9 Example traversal of GBPlanner + Frontier SS. At startup local diffusion is run to allow for immediate planning. Once a graph is constructed and a path is planned the robot begins both traversing the path and making predictions of the frontier occupancy (where predictions are shown in grey). Upon reaching its destination, the graph is extended based on the observed and predicted occupancy and the robot can continue exploring

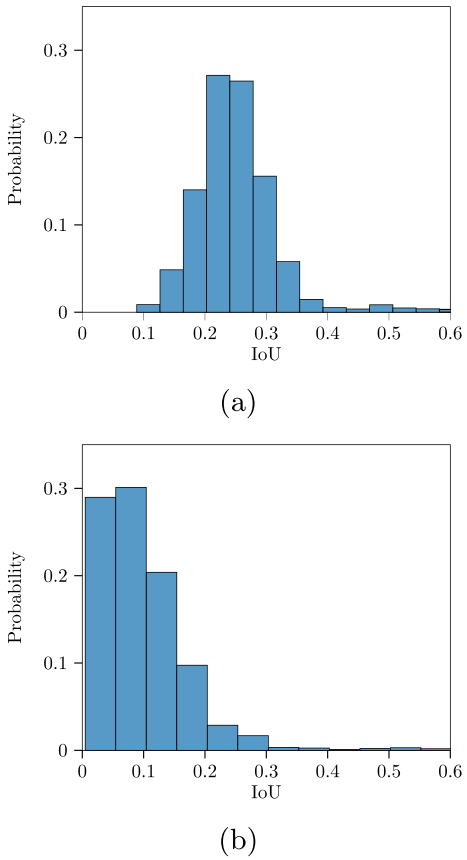


Fig. 10 Env 2 IoU Probability Mass Function (PMF). **a** IoU histogram of RC SceneSense predictions. **b** IoU histogram of FC SceneSense predictions. The IoU distributions show that RC occupancy predictions are more likely to be similar than FC predictions

Interestingly, The KID values are nearly identical between SS-RC-PMM and SS-FC-PMM, indicating that the model occupancy predictions at range are as reasonable as the predictions made around the robot, even though there is less information for the predictions at range. KID is known to be less sensitive to outliers when compared to FID (Bińkowski et al., 2018b). The unreasonable predictions that can occur when performing FC predictions are likely better filtered out by the KID metric, resulting in similar scores.

The large discrepancy between OSMM and PMM results when evaluated at frontiers is due to the sparsity of the occupancy map at the frontier. We predict that as the number of unknown voxels grow, so too does the distribution of predicted scenes. Intuitively, if there are no observed voxels to guide the prediction SceneSense will predict a wide variety of possible occupancy maps. Conversely, if all voxels in the target space are observed, the same occupancy map will be generated every time.

We can analyze the number of available voxels for occupancy prediction as the number of unknown voxels in the target area x_{rm} as a percentage of the total observed voxels in x_{gt}. Using the results from environment 2 to evaluate this prediction, we calculate that on average 59.18% of target area voxels are unknown when performing RC occupancy prediction. However, when performing FC prediction this number jumps to 70.79%. This result supports the intuitive statement that there are more available (unknown) voxels for prediction around the frontiers of the map than around the robot.

To confirm that the increase in unknown voxels widens the distribution of occupancy prediction we generate a distribution of results by calculating the IoU of each prediction against all other predictions made during the run. The results of this approach as provided in Figure 10 show that RC predictions are more likely to be similar, while FC predictions are more likely to be dissimilar with very little overlap. When predictions are all similar, PMM becomes less important for accurate predictions, since OSMM would result in a similar map each time. However PMM is needed

at range to achieve reasonable results since it can negotiate the wider distribution of possible occupancy predictions.

We see an expected increase in FID and KID for frontier-centric prediction. Around the robot, more of the space has been directly observed with lidar sensor measurements. This means less unknown voxels are available for generative prediction since SceneSense does not override occupancy values for voxels that are observed occupied or free based on sensor measurements. As a result, the underlying structure of the environment is more similar between predictions and the ground truth. This results in lower FID and KID values for robot-centric. When making predictions at the frontier, more unknown voxels are available, leading to larger variance in the predictions. This can result in a larger difference between the ground truth map and the predicted occupancy generated with SceneSense, leading to larger FID and KID values.

8 Extended exploration experiments

In this section we examine SceneSense as a "drop-in" method for autonomous exploration. First we examine key exploration scenarios where SceneSense solves existing map based exploration problems and evaluate the systems performance at autonomously navigating narrow corners. Then, we evaluate the robots exploration performance on full scene explorations, in 2 dissimilar environments.

8.1 Solved robotic exploration tasks

Robotic systems often exhibit poor or unpredictable performance in edge cases, particularly when encountering rare or unforeseen scenarios. Many edge cases in robotics are known, and need special error handling processes to circumvent errors when in these configurations. With the addition of SceneSense to our robotic framework we alleviate the need for special error handling for some common edge cases seen during deployment. In particular, the common sense occupancy filling alleviates cases where robotic systems cannot generate plans over ares in which they cannot directly observe although it would be common sense that a traversable path exists. We provide two examples of configurations where planning failed without the probabilistic map merging and one example where traditional terrain filling methods would fail where SceneSense does not.

8.1.1 Start up

The first configuration where the robotic planner fails in during start up. Given the mounting point of the lidar and height of the robot, there are several meters around the platform that cannot be observed directly by the lidar. At startup this results in a large hole around the platform, which the planner sees as untraversable terrain and therefore is unable to form a plan for traversable as shown in Fig. 5. Algorithmic hole filling methods attempt to solve this problem but these methods result in unexpected failures when traversing new environments and require online tuning for fill size. Furthermore, these methods require specific handling of the "start" configuration requiring additional logic for the robot exploration method. A different approach is to add additional depth sensors intended to point directly at what the robot is walking on. However this requires the purchase of additional sensors which can be prohibitively expensive, and would require additional ongoing calibration between the sensor to ensure alignment of measurements. Alternatively, SceneSense is able to fill the hole at startup and immediately begin planning, without additional sensors or special configuration handling.

8.1.2 Narrow hallways and corners

During Traversal, perception systems can miss measurements of necessary information to continue exploration. In our case, due the the high mounting point of our lidar sensor, traversal of narrow hallways can lead to missing ground measurements necessary to continue exploration as shown in Fig. 6a. In some sampling based planners for ground systems, it is not allowed to sample points above open space, and therefore the robot will never traverse down the hallway to continue exploration. SceneSense provides realistic filling in this scenario and allows the planner to continue exploration down the hallway.

We can directly measure the impact of SceneSense on robot navigation in this particular scenario. We select a L-shaped test course as shown in Fig. 8 (which was not included in the training set) where the robot must make a narrow 90- degree turn. We evaluate the robot performance as its time to complete the course with the following 4 different configurations:

1. *User Operation Ground Truth*: The system is teleoperated by the user to generate a ground truth result for human-like exploration.
2. *GBPlanner* (Dang et al., 2020): Baseline GBPlanner is used to explore the narrow hallway environments.
3. *GBPlanner + Local SceneSense* : Baseline GBPlanner is used to explore the narrow hallway environments. SceneSense predictions are merged with the running occupancy map as the system explores. SceneSense only makes local occupancy predictions (Predictions centered around the robot).

Table 2 Single Hallway Results: Single hallway traversal results averaged over 8 runs per configuration. All time measurements are in seconds, and distance measurements are in meters

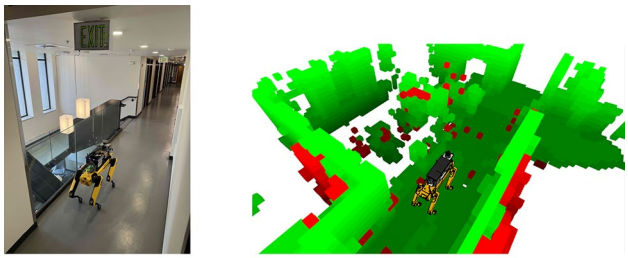
Method	Exp. time (s)	Min time (s)	Max time (s)	σ_{exp}(s)	Fail count
Human Operator	9.35	9.00	9.70	0.34	0
GB (Dang et al., 2020)	25.98	18.33	33.77	5.18	1
GB + Local SS	23.77	18.48	29.27	3.81	0
GB + Frontier SS	21.10	16.35	28.17	3.53	0

4. *GBPlanner + Frontier SceneSense* : Baseline Planner is used to explore the narrow hallway environments. SceneSense predictions are merged with the running occupancy map as the system explores. SceneSense makes predictions at range for 1 target point in the graph with the highest exploration gain.

8.1.3 Discussion

As shown in Table 2, enhancing the running occupancy map using the frontier SceneSense predictions results in the best autonomous performance of the system. The performance of GBPlanner + Frontier SS is not only the lowest mean exploration time and path length, but also has a lower variance than the other methods. We believe that the primary reason that frontier SS outperforms local SS is due to system latency. In particular the latency introduced in running multiple diffusion and merging those predictions into the map. As shown in Fig. 9 the robot is able to perform 4 consecutive occupancy predictions while traversing the calculated path. Each of these predictions takes approximately 0.5 seconds to generate and similar time to merge with the running map. That means that by the time that the robot has made it to the area where it will need to plan its next path the SceneSense predictions will have already been merged into the map for quick planning. When performing local occupancy prediction its takes time to to make the occupancy predictions and update the running occupancy map, resulting in the robot frequently pausing at the end of its initial path. If this prediction and update latency was reduced the local SceneSense predictions should perform just as well as the frontier predictions.

Importantly, a traversability failure is observed in this experiment when operating over the sensor only map. This failure is observed when the robot traverses particularly close to the right wall of the intersection, and therefore cannot observe the ground necessary for traversal. A similar observation problem is show in Fig. 6. The SceneSense enhanced map is able to make this logical inference of ground existence every time in our testing, allowing for more reliable exploration.



(a) (b)

Fig. 7 **a** The robot's position during exploration near a glass railing, an adversarial scenario for planners that do not consider ground traversability. **b** The running occupancy map shows that the ground suddenly ends to the right of the robot, where the railing prevents it from navigating off the edge. Samplers that do not check for traversability–effective in solving scenarios in Figs. 6 and 5–could mistakenly allow paths through open space. Here, SceneSense correctly generates red voxels to indicate untraversable ground, preventing unsafe navigation

8.1.4 SceneSense robustness to adversarial scenarios

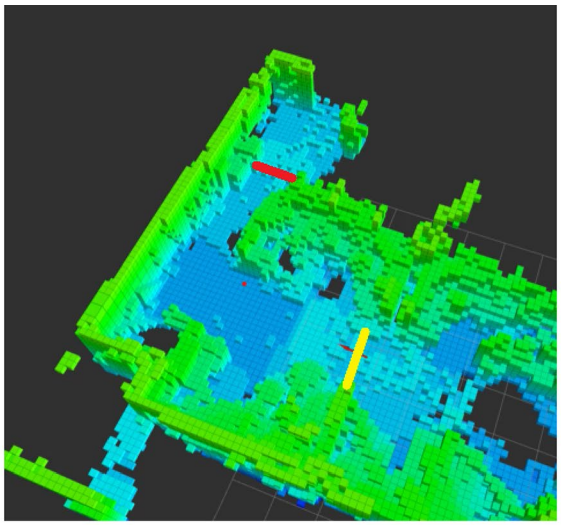
While promising, the results presented in Sects. 8.1.1 and 8.1.2 could be solved with simple brute force solution such as removing the ground checking from the graph sampler. To further differentiate SceneSense from potential brute force solutions we identify adversarial scenarios where areas of the map are not traversable. One such scenario is shown in Fig. 7, where the robot traverses near a glass railing unobservable by the lidar sensor. While a naive unconstrained sampling approach could sample points and paths above the empty space, SceneSense is able to differentiate this space from other potential ground areas and not predict ground over the edge.

8.2 Full robot exploration

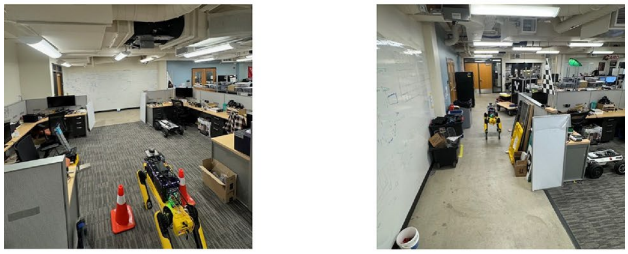
In this section we explore the end effects of SceneSense when applied to a robotic exploration platform. For these experiments SceneSense is considered a "drop-in" method that does not require custom planning or mapping frameworks.

8.2.1 Experimental setup

To evaluate the exploration speed of the system we deployed the platform in test environments shown in Fig. 11a,b. The robot is started once at each position indicated by the black boxes in the associated figures (5 positions in environment 1, 4 positions in environment 2). One run per starting position is then run per configuration, and the results are averaged to generate Fig. 13 and Tables 3 and 4. The same testing configurations described in Sect. 8.1.2 are used for all runs.

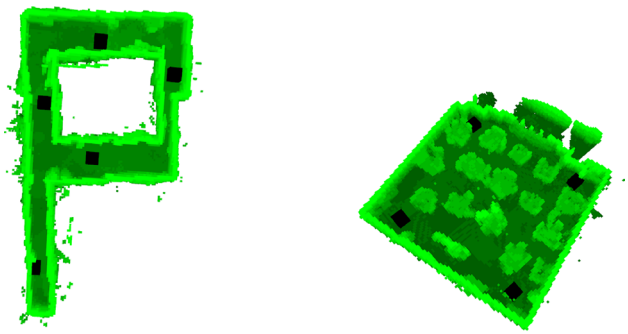


(a)



(b)

Fig. 8 Test Environment: The test environment used to produce Table 2. **a** The complete occupancy map representation with the start and finish line shown in yellow and red, respectively. **b** The starting position (left) and finish position (right) of the Spot robot



(a) (b)

Fig. 11 Two test environments for exploration evaluation. The green voxels represent the complete occupancy map, and the black cubes indicate the robot's starting positions. **a** Environment 1, **b** Environment 2

Table 3 Env. 1 Exploration Results: Results from exploration of the environment shown in Fig. 11a. All results are calculated based on reaching 95% environment exploration

Method	Exp. time (s)	Min time (s)	Max time (s)	σ_{exp} (s)	$\frac{\text{exp\%}}{\text{sec}}$
Human Operator	64.03	36.75	75.05	8.717	1.327
GB (Dang et al., 2020)	152.27	57.75	148.45	13.027	0.529
GB + Local SS	151.83	91.65	141.99	9.629	0.565
GB + Frontier SS	N.A	N.A	N.A	7.86	N.A

Table 4 Env. 2 Exploration Results: Results from exploration of the environment shown in Fig. 11b. All results are calculated based on reaching 95% environment exploration

Method	Exp. time	Min time	Max time	σ_{exp}	$\frac{\text{exp\%}}{\text{sec}}$
Human Operator	55.76	40.95	62.55	10.445	1.516
GB (Dang et al., 2020)	106.46	69.68	117.55	5.394	0.624
GB + Local SS	81.68	74.35	91.05	5.685	0.866
GB + Frontier SS	N.A	N.A	N.A	6.993	N.A

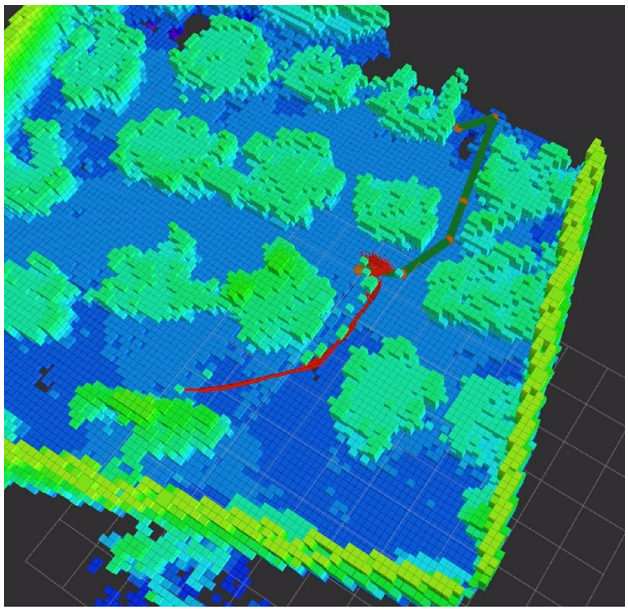


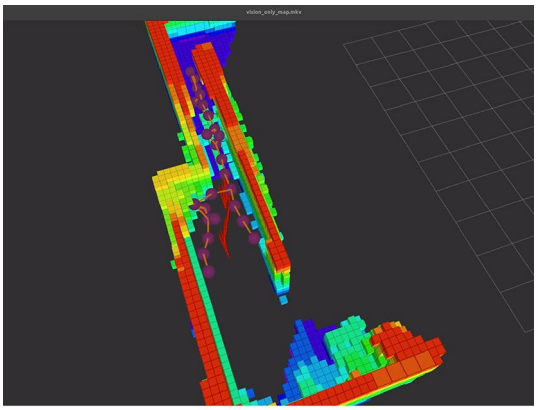
Fig. 14 Unintuitive planning. As the robot is traversing the classroom environment, a hole in the map leads to a loop-back path. The robot odometry is shown in red, and the planned path is shown in green. Due to the hole in the map, the robot's plan configures it into a poor position to continue exploration

8.2.2 Discussion

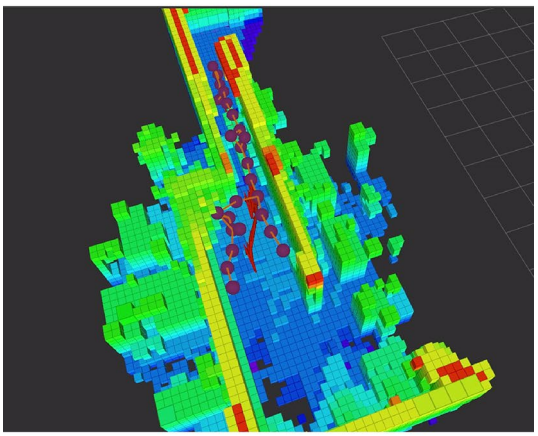
Our experiments demonstrate that SceneSense-enhanced maps consistently improve exploration performance compared to vision only maps. In both test environments, SceneSense yields lower average and maximum exploration times with higher exploration gain per second. The vision only maps showed faster minimum exploration times, but with

much larger maximum exploration times, indicating far greater variance. These results suggest lower consistency of operation when relying solely on vision. In some starting configurations the robot may make observations that enable rapid exploration, while in others it may make poor observations that increase total exploration time. Local occupancy prediction narrows this distribution, resulting in a system that is less dependent on a favorable direct observation at startup.

Our data revealed that startup configurations lead to some environment specific behavior. The minimum time in environment 1 is attributed to the vision-only GBPlanner navigating straight down the hallway when starting from the dead end, instead of turning right at the first junction as seen in Fig. 11a. With the more complete ground representation from the SceneSense enhanced map, GBPlanner instead navigates down the larger hallway to the right first, which increases its overall completion time. Without this outlier in the environment 2 data collection, the exploration rate of the SceneSense enhanced planner increases by 32%. It is our hypothesis that the substantial exploration rate gains are actually attributed to the startup filling. As shown in Fig. 12b, the startup filling has a lingering effect on the map even after some initial exploration. It is likely that the exploration gain in these areas remains significant, resulting in the planner choosing to double back to explore the starting position in several instances.



(a)



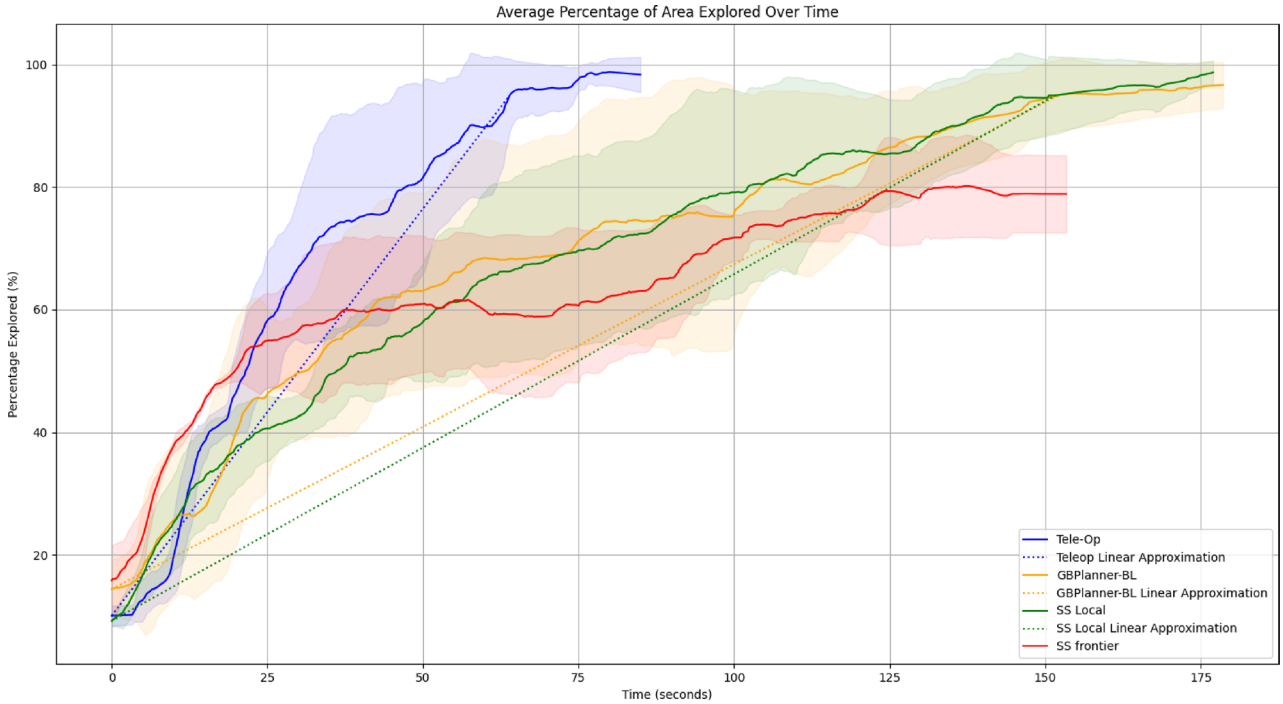
(b)

Fig. 12 **a** The vision only map. **b** The SceneSense enhanced map. The current planning graph is shown as the purple and red graph structures. In (**a**), the vision only map contains a large gap near the robot's starting position. Due to occlusions from the narrow hallway, the hole is not observed and therefore remains unpopulated. SceneSense, however, merges its predictions with the existing map to make a realistic estimation of the unobserved space, filling the holes and allowing the robot to plan over the environment. Traversable occlusions, such as vertical structures, are correctly negotiated out of the map when the robot makes observations of the space

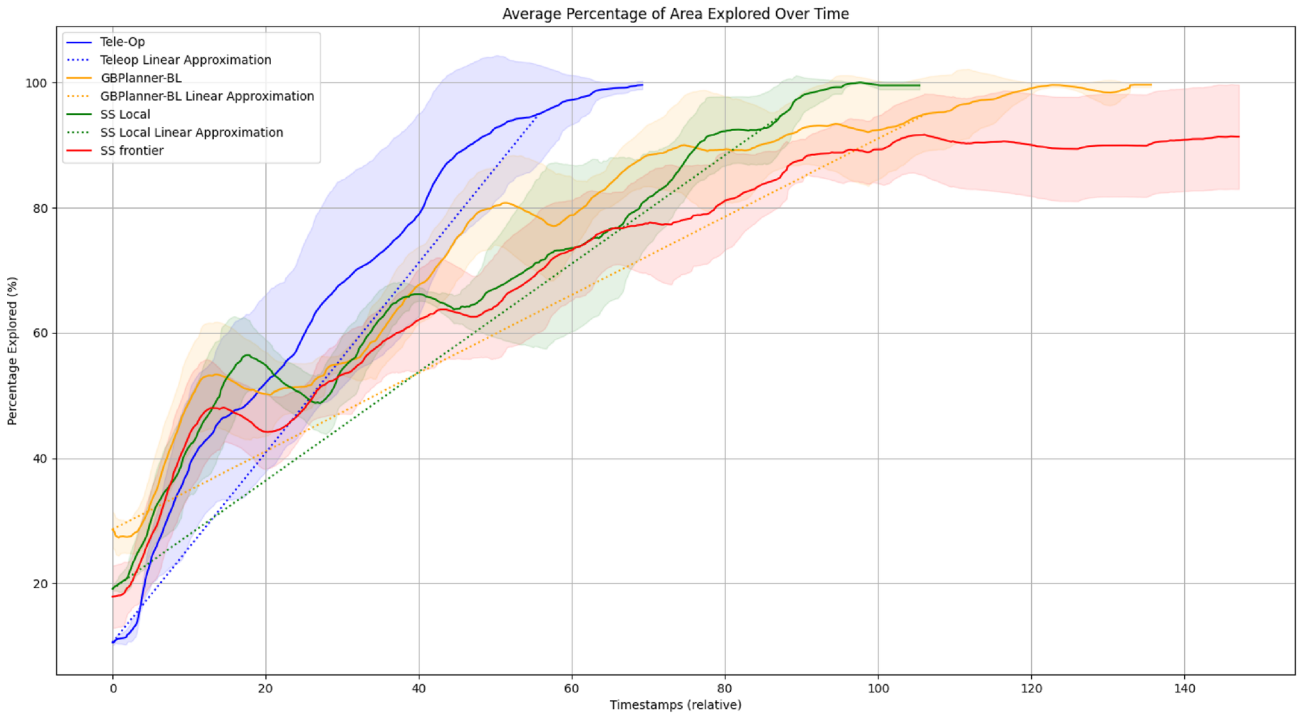
We observed a clear distinction between local SceneSense predictions and frontier-centric SceneSense predictions. Local predictions improved planning performance both by filling in startup regions and by eliminating small holes in the map that otherwise can cause unintuitive behavior. For instance, in Fig. 14, a small hole causes the planner to loop back on itself rather than taking a direct path. SceneSense fills such gaps, producing maps that enable coherent trajectory generation and thereby contributing to faster and more consistent exploration.

In contrast, frontier-centric SceneSense (SS frontier), while initially promising, was unable to complete exploration as seen in Fig. 13. This failure arises because diffusion predictions overwrite unobserved voxels with occupancy probability values, which erases the boundary between free and unknown that frontier planners rely on. As a result, the planner prematurely concludes that the environment is fully explored. One possible mitigation is to add an indicator variable that marks whether a voxel was generated by diffusion or observed directly. While this allows for differentiation between diffused predicted voxels and sensor observed voxels, it does not recover the geometry of the frontier and would require the planner to inherently understand when to reason over flag based voxels and when to reason over the traditional sensor observed voxel representations. In practice, the planner would need to reason explicitly over the interface between diffusion-generated occupancies and those derived from sensor measurements in order to continue exploration using frontier based exploration. These results suggest that new planners are needed which can reason jointly about observed and predicted occupancy, preserving the structural cues necessary for frontier-based exploration.

Although SS frontier did not complete exploration, we argue it remains a valid and motivating contribution. First, it demonstrates the potential of extending SceneSense beyond local completions toward map-wide inference. Second, its failure case highlights the broader need for exploration planning paradigms that explicitly incorporate predictive



(a)



(b)

Fig. 13 **a** Results associated with runs in Env. 1, as shown in Fig. 11a. **b** Results associated with runs in Env. 2, as shown in Fig. 11b. The robot starts once at each of the positions in Fig. 11a, b . The average exploration percentage is shown for each configuration, where the shaded region represents the standard deviation across runs at that time step. Additionally, linear approximations illustrate the average exploration rate required to achieve 95% coverage of the test environment

models with joint reasoning, rather than treating predictions as direct drop in substitutions for perception. These insights point toward important research questions on integrating predicted occupancy into higher level reasoning and planning systems, which pursue in future work.

9 Limitations

Despite the promising results demonstrated by our proposed method for generative occupancy map synthesis, several limitations warrant discussion and present opportunities for future work.

Currently, training and evaluation of SceneSense (SS) is limited to structured indoor environments. The generalization of our method to more diverse, complex scenarios, such as outdoor, unstructured, or highly dynamic environments, remains an open question. The training data, collected from a single building, may not fully capture the variability present in real-world conditions. Future work could train the model on broader datasets to improve its robustness and generalizability.

Our system's reliance on an off-board GPU for inference presents a practical limitation for autonomous systems with strict power and size constraints. Deployment on embedded platforms would require model optimization and potentially an exploration of more efficient architectures.

Incorrect predictions, particularly in scenarios involving transparent or reflective surfaces, could produce unpredictable behavior. While the probabilistic map merging (PMM) framework helps mitigate the impact of some erroneous predictions by favoring sensor measurements, a single, highly confident but incorrect prediction can still adversely affect the map's quality and the robot's subsequent planning.

Finally, while our approach improved map quality and traversability, it did not always result in a complete exploration of the environment, as seen in the case of the SS-frontier approach (Fig. 13). This suggests a potential mismatch between the current frontier detection algorithms, which may be confused by the predicted occupancy values, and the generative map. Future work addressing frontier exploration methods for generative occupancy could ensure that the robot can successfully complete exploration.

10 Conclusion

In this work we have presented the general occupancy mapping tool, SceneSense, implemented the model onboard a real-world system, and evaluated both the realism of the predictions as well as the predictions impacts on robotic exploration. We show that the SceneSense model is able to enhance existing occupancy mapping methods and use those hybrid maps as a basis for more constant exploration of space. Further we show that the integration of SceneSense into existing planning frameworks enables the planner to traverse environments that are traditionally challenging to complete with only direct observations, such as narrow hallway traversal or directly after robot startup. Future work includes the design and implementation of planners that can distinguish predicted and observed space, and balance decision making based on observations made in both modalities. Furthermore the SceneSense model can be retrained and scaled up to account for more diverse environments using readily available online datasets in a foundation model style approach. We present SceneSense as an empirically validated approach for enhancing existing robotic planning stacks by providing a "drop-in" method that allows for robots to make common-sense inference of unobserved geometry.

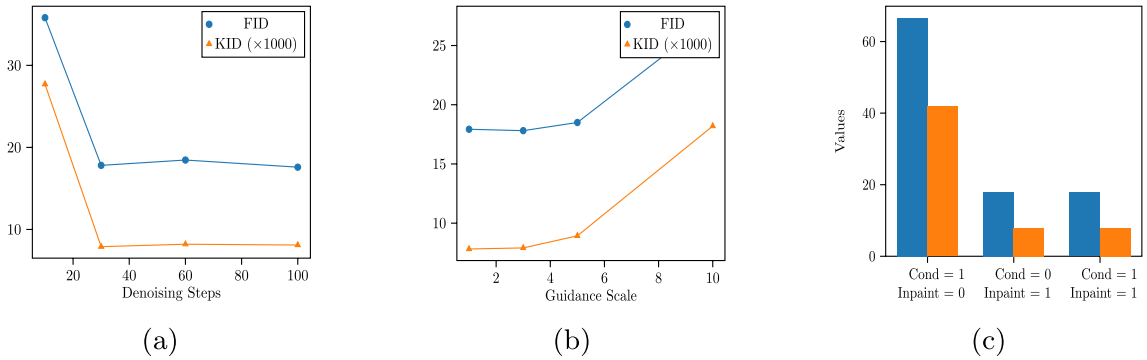


Fig. 15 SceneSense Ablation Experiments. All ablation experiments were conducted on Test House 1 using the same trained diffusion network. For all experiments, conditioning and inpainting were enabled, s was set to 3, and 30 denoising steps were used unless these values were being ablated. **a** Evaluation of various denoising step values. **b** Evaluation of different guidance scale s values, further discussed in 10. **c** Ablation of conditioning and inpainting for the network, where 1 indicates enabled and 0 indicates disabled

Appendix A: SceneSense ablation study

This ablation study was first presented in Reed et al. (2024) where the SceneSense occupancy prediction network was a conditional diffusion model (as opposed to the unconditional model presented here). When examining these results it was concluded that in this particular application the conditional network does not provided substantial benefits over the equivalent unconditional network.

Denoising steps discussion

The number of diffusion steps defines the size of each diffusion step during the reverse diffusion process. Generating reasonable results using the fewest possible denoising steps is desirable behavior to reduce computation time. Additionally too many denoising steps have been shown to introduce sampling drift which results in decreased performance (Chen et al., 2022; Ji et al., 2023). As shown in Fig. 15a our method saw the best results when configured to 30 denoising steps. Too few denoising steps results in the network being unable make accurate predictions over the large time step. Increasing the number of denoising steps keeps results relatively stable over time, however you can see the KID is slightly worse using more steps due to sampling drift. Sampling drift is a result of the discrepancy between the distribution of the training and the inference data. During training, the model is trained to reduce a noisy map x_t to a ground truth map x, at inference time the model iteratively removes noise from its already imperfect noise predictions. These predictions will drift away from the initial corruption distribution which becomes more pronounced at smaller time steps due to the compounding error.

Table 5 Inference time and model size results. "Full inference" and "end-to-end" evaluations are computed using 30 diffusion steps

	Cond. model (Reed et al., 2024)	Uncond. model
Trainable params	141,125,261	101,144,845
Diffusion step (s)	0.03707	0.0147
Full inference (s)	1.11	0.4437
Backbone (s)	0.55099	N.A
End-to-End (s)	1.66	0.4437

Conditioning and guidance scale discussion

The guidance scale s as defined in Ho et al. (2020) is a constant that multiplies the difference of the conditional diffusion and unconditional diffusion to "push" the diffusion process towards the conditioned answer. Setting s too high results in too large of pushes away from reasonable predictions and results in poor generalization to new environments. The best FID is measured when $s = 3$ however KID is slightly lower when $s = 1$. Further, when examining chart (c) it is shown that the results when conditioning is removed all together ($s = 0$) are very similar to the best results seen with conditioning enabled (albeit slightly worse). This is likely because most of the useful conditioning information is captured in the local occupancy data, and mapping measured RGB-D points from areas in front of the geometry to under or behind the platform is a very difficult task. The performance of the conditioning only data may be seen to have a larger impact on the overall results if the sensor could capture more local information, such as given a wider FOV or different mounting angle.

Appendix B: Conditional versus unconditional diffusion

This ablation study was first presented in Reed et al. (2025) to evaluate the inference time gains of removing cross-attention based conditioning from the SceneSense occupancy prediction network.

In this ablation we implement the same denoising network structure presented in Reed et al. (2024). It is a U-net constructed from the HuggingFace Diffusers library of blocks (Platen et al., 2022) and consists of Resnet (He et al., 2015) downsampling/upsampling blocks. The diffusion model is trained using randomly shuffled pairs of ground truth local occupancy maps x. We use Chameleon cloud computing resources (Keahey et al., 2020) to train our model on one A100 with a batch size of 32 for 250 epochs or 88, 208 training steps. We use a cosine learning rate scheduler with a 500 step warm up from 10^{-6} to 10^{-4}. The noise scheduler for diffusion is set to 1, 000 noise steps.

At inference time we evaluate our dataset using an RTX 4070 TI Super GPU for acceleration. The number of diffusion steps is configured to 30 steps.

Inference time

We evaluate the inference time of the unconditional diffusion model against the inference time of the conditional model presented in the original SceneSense paper (Reed et al., 2024). The cross-attention enabling trainable parameters are removed for the unconditional model, but the number of output channels for the constructed U-net are held constant between both models. As the ablation results of the original paper show minor, or no performance gain between the conditional and unconditional model in this configuration we do not evaluate the results of the model predictions in these experiments.

As shown in Table 5, removing the conditioning from the diffusion model reduces the computation requirements substantially. The unconditioned model reduces the number of trainable parameters by 28%, the model inference time by 60% and the end to end computation time by 73%. These improvements enable SceneSense to operate in real-time more effectively, allowing for more flexible implementations for onboard robotic applications.

Author contributions L.A. and A.R. wrote the main manuscript. B.C. prepared equations and figures. All authors reviewed the manuscript.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Ahmad, S., Mills, A. B., Rush, E. R., Frew, E. W., & Humbert, J. S. (2021). 3d reactive control and frontier-based exploration for unstructured environments. In: 2021 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 2289–2296). IEEE.

Ahmad, S., Sunberg, Z. N., & Humbert, J. S. (2021). Apf-pf: Probabilistic depth perception for 3d reactive obstacle avoidance. In: 2021 American control conference (ACC) (pp. 32–39). IEEE.

Bernard, M., Kondak, K., Maza, I., & Ollero, A. (2011). Autonomous transportation and deployment with aerial robots for search and rescue missions. *Journal of Field Robotics, 28*(6), 914–931.

Biggie, H., Rush, E. R., Riley, D. G., Ahmad, S., Ohradzansky, M. T., Harlow, K., Miles, M. J., Torres, D., McGuire, S., Frew, E. W., et al. (2023). Flexible supervised autonomy for exploration in subterranean environments. arXiv preprint arXiv:2301.00771

Bińkowski, M., Sutherland, D. J., Arbel, M., & Gretton, A. (2018). Demystifying mmd gans. arXiv preprint arXiv:1801.01401

Bińkowski, M., Sutherland, D. J., Arbel, M., & Gretton, A. (2018). Demystifying MMD GANs. In: International conference on learning representations. https://openreview.net/forum?id=r1lUOzWCW

Carvalho, J., Le, A. T., Baierl, M., Koert, D., & Peters, J. (2023). Motion planning diffusion: Learning and planning of robot motions with diffusion models. In: 2023 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 1916–1923). https://doi.org/10.1109/IROS55552.2023.10342382

Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., & Zhang, Y. (2017). Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158

Chen, S., Sun, P., Song, Y., & Luo, P. (2022). Diffusiondet: Diffusion model for object detection. arXiv preprint arXiv:2211.09788

Chung, T. H., Orekhov, V., & Maio, A. (2023). Into the robotic depths: Analysis and insights from the darpa subterranean challenge. *Annual Review of Control, Robotics, and Autonomous Systems, 6*, 477–502.

Dang, T., Tranzatto, M., Khattak, S., Mascarich, F., Alexis, K., & Hutter, M. (2020). Graph-based subterranean exploration path planning using aerial and legged robots. *Journal of Field Robotics, 37*(8), 1363–1388.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., & Sutskever, I. (2020). Jukebox: A generative model for music. arXiv preprint arXiv:2005.00341

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision, 88*(2), 303–338.

Freda, L., & Oriolo, G. (2005). Frontier-based probabilistic strategies for sensor-based exploration. In: Proceedings of the 2005 IEEE international conference on robotics and automation (pp. 3881–3887). IEEE.

González-Banos, H. H., & Latombe, J.-C. (2002). Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research, 21*(10–11), 829–848.

Harvey, W., Naderiparizi, S., Masrani, V., Weilbach, C., & Wood, F. (2022). Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems, 35*, 27953–27965.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems *30*.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems, 33*, 6840–6851.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*. https://doi.org/10.1007/s10514-012-9321-0. Software available at https://octomap.github.io

Huang, R., Zhao, Z., Liu, H., Liu, J., Cui, C., & Ren, Y. (2022). Prodiff: Progressive fast diffusion model for high-quality text-to-speech. In: Proceedings of the 30th ACM international conference on multimedia (pp. 2595–2605).

Huang, Y., Zheng, W., Zhang, B., Zhou, J., & Lu, J. (2024). Selfocc: Self-supervised vision-based 3d occupancy prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) (pp. 19946–19956).

Ji, Y., Chen, Z., Xie, E., Hong, L., Liu, X., Liu, Z., Lu, T., Li, Z., & Luo, P. (2023). Ddp: Diffusion model for dense visual prediction. arXiv e-prints, 2303.

Kapelyukh, I., Vosylius, V., & Johns, E. (2023). Dall-e-bot: Introducing web-scale diffusion models to robotics. *IEEE Robotics and Automation Letters, 8*(7), 3956–3963. https://doi.org/10.1109/LRA.2023.3272516

Kawatsuma, S., Fukushima, M., & Okada, T. (2012). Emergency response by robots to fukushima-daiichi accident: Summary and lessons learned. *Industrial Robot: An International Journal, 39*(5), 428–435.

Keahey, K., Anderson, J., Zhen, Z., Riteau, P., Ruth, P., Stanzione, D., Cevik, M., Colleran, J., Gunawi, H. S., Hammock, C., Mambretti, J., Barnes, A., Halbach, F., Rocha, A., & Stubbs, J. (2020). Lessons learned from the chameleon testbed. In: Proceedings of the 2020 USENIX annual technical conference (USENIX ATC '20). USENIX Association.

Keidar, M., & Kaminka, G. A. (2014). Efficient frontier detection for robot exploration. *The International Journal of Robotics Research, 33*(2), 215–236.

Kim, S. W., Brown, B., Yin, K., Kreis, K., Schwarz, K., Li, D., Rombach, R., Torralba, A., & Fidler, S. (2023). Neuralfield-ldm: Scene generation with hierarchical latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8496–8506).

Kruijff-Korbayová, I., Grafe, R., Heidemann, N., Berrang, A., Hussung, C., Willms, C., Fettke, P., Beul, M., Quenzel, J., Schleich, D., et al. (2021). German rescue robotics center (drz): A holistic approach for robotic systems assisting in emergency response. In: 2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR) (pp. 138–145). IEEE.

Luo, S., & Hu, W. (2021). Diffusion probabilistic models for 3d point cloud generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2837–2845).

Ma, W. -C., Tartavull, I., Bârsan, I. A., Wang, S., Bai, M., Mattyus, G., Homayounfar, N., Lakshmikanth, S. K., Pokrovsky, A., & Urtasun, R. (2019). Exploiting sparse semantic hd maps for self-driving vehicle localization. In: 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 5304–5311). IEEE.

Makarenko, A. A., Williams, S. B., Bourgault, F., & Durrant-Whyte, H. F. (2002). An experiment in integrated exploration. In: IEEE/RSJ international conference on intelligent robots and systems (vol. 1, pp. 534–539). IEEE.

Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J. -Y., & Ermon, S. (2022). SDEdit: Guided image synthesis and editing with stochastic differential equations. In: International conference on learning representations.

Naeem, M. F., Oh, S. J., Uh, Y., Choi, Y., & Yoo, J. (2020). Reliable Fidelity and Diversity Metrics for Generative Models.

Nichol, A. Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., Mcgrew, B., Sutskever, I., & Chen, M. (2022). GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th international conference on machine learning. proceedings of machine learning research (vol. 162, pp. 16784–16804). PMLR, https://proceedings.mlr.press/v162/nichol22a.html

Park, S., Oh, Y., & Hong, D. (2017). Disaster response and recovery from the perspective of robotics. *International Journal of Precision Engineering and Manufacturing, 18*, 1475–1482.

Parmar, G., Zhang, R., & Zhu, J. -Y. (2022). On aliased resizing and surprising subtleties in gan evaluation. In: CVPR.

Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., & Wolf, T. (2022). Diffusers: State-of-the-art diffusion models. GitHub.

Reed, A., Achey, L., Crowe, B., Hayes, B., & Heckman, C. (2025). Online diffusion-based 3d occupancy prediction at the frontier with probabilistic map reconciliation. In: 2025 IEEE international conference on robotics and automation (ICRA) (pp. 2846–2852). https://doi.org/10.1109/ICRA55743.2025.11128832

Reed, A., Crowe, B., Albin, D., Achey, L., Hayes, B., & Heckman, C. (2024). Scenesense: Diffusion models for 3d occupancy synthesis from partial observation. In: 2024 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 7383–7390). https://doi.org/10.1109/IROS58592.2024.10802589

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10684–10695).

Schwarz, M., Rodehutskors, T., Droeschel, D., Beul, M., Schreiber, M., Araslanov, N., Ivanov, I., Lenz, C., Razlaw, J., Schüller, S., et al. (2017). Nimbro rescue: Solving disaster-response tasks with the mobile manipulation robot momaro. *Journal of Field Robotics, 34*(2), 400–425.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., & Rus, D. (2020). LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning (pp. 2256–2265). PMLR.

Tang, J., Nie, Y., Markhasin, L., Dai, A., Thies, J., & Nießner, M. (2023). Diffuscene: Scene graph denoising diffusion probabilistic model for generative indoor scene synthesis. arXiv preprint arXiv:2303.14207

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. -A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971

Tranzatto, M., Dharmadhikari, M., Bernreiter, L., Camurri, M., Khattak, S., Mascarich, F., Pfreundschuh, P., Wisth, D., Zimmermann, S., Kulkarni, M., Reijgwart, V., Casseau, B., Homberger, T., Petris, P. D., Ott, L., Tubby, W., Waibel, G., Nguyen, H., Cadena, C., Buchanan, R., Wellhausen, L., Khedekar, N., Andersson, O., Zhang, L., Miki, T., Dang, T., Mattamala, M., Montenegro, M., Meyer, K., Wu, X., Briod, A., Mueller, M., Fallon, M., Siegwart, R., Hutter, M., & Alexis, K. (2022). Team CERBERUS Wins the DARPA Subterranean Challenge: Technical Overview and Lessons Learned.

Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K., et al. (2022). Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems, 35*, 10021–10039.

Verma, V., Nash, J., Saldyt, L., Dwight, Q., Wang, H., Myint, S., Biesiadecki, J., Maimone, M., Tumbar, A., Ansar, A., Kubiak, G., & Hogg, R. (2024). Enabling long & precise drives for the perseverance mars rover via onboard global localization. In: 2024 IEEE aerospace conference (pp. 1–18). https://doi.org/10.1109/AERO58975.2024.10521160

Wang, L., Ye, H., Wang, Q., Gao, Y., Xu, C., & Gao, F. (2021). Learning-based 3d occupancy prediction for autonomous navigation in occluded environments. In: 2021 IEEE/RSJ International conference on intelligent robots and systems (IROS) (pp. 4509–4516). IEEE.

Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In: Proceedings 1997 IEEE international symposium on computational intelligence in robotics and automation CIRA'97.'Towards New Computational Principles for Robotics and Automation' (pp. 146–151). IEEE.

Yamauchi, B., Schultz, A., & Adams, W. (1998). Mobile robot exploration and map-building with continuous localization. In: Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146) (vol. 4, pp. 3715–3720). IEEE.

Yu, S., Sun, F., Guo, J., Zhang, R., & Cheng, X. (2022). Legonet: A fast and exact unlearning architecture. arXiv preprint arXiv:2210.16023

Yuan, K., Sajid, N., Friston, K., & Li, Z. (2023). Hierarchical generative modelling for autonomous robots. *Nature Machine Intelligence, 5*(12), 1402–1414.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Lorin Achey is a current Ph.D. student in Computer Science at the University of Colorado Boulder and a recipient of the National Defense Science Engineering Graduate (NDSEG) Fellowship. Ms. Achey received her B.S. in Computer Science from the Georgia Institute of Technology. Prior to returning to graduate school, she worked as a Software Engineer for Qualcomm, Inc. and Amazon Web Services. Ms. Achey's research interests lie in multimodal generative world models, 3D scene understanding, and embodied AI for mobile robotics. Ms. Achey develops algorithms that combine vision, 3D geometric reasoning, diffusion-based generative modeling, and vision–language–action models to help robots form predictive, spatially grounded representations of their environment.



Alec Reed received his M.Sc. in Electrical Engineering from the University of Washington in 2019 and his Ph.D. in Computer Science from the University of Colorado, Boulder in 2024. Dr. Reed is currently a Senior Roboticist at Overland AI where he works on real world robot applications of Generative AI. Dr. Reed's research interests lie in the applications of generative AI and shared embeddings spaces to enhance real-world robots.



Brendan Crowe is a Ph.D. Student in Computer Science at the University of Colorado Boulder, Boulder, CO, USA. Mr. Crowe received his BS in Statistics from the College of Engineering and Physical Sciences, University of New Hampshire, Durham, NH, USA, in 2021. Mr. Crowe's research interests include semantic scene completion and robotic mapping using generative methods, as well as millimeter-wave radar



Bradley Hayes is an Associate Professor at the University of Colorado Boulder. Dr. Hayes' work develops techniques to create autonomous robots that are capable teammates who learn from and collaborate with humans, making people more efficient, capable and safe during task execution. This research combines novel innovations in machine learning and explainable artificial intelligence, providing robots with the ability to learn and generalize skills without programming and to make human-robot teams more powerful than the sums of their parts. Prior to joining the CU Boulder faculty, Bradley Hayes spent eight years conducting research on the algorithmic foundations of human-robot interaction at the Yale Social Robotics Lab and the Massachusetts Institute of Technology Interactive Robotics Group.



Christoffer Heckman is an Associate Professor in the Department of Computer Science at the University of Colorado at Boulder and the Jacques I. Pankove Faculty Fellow in the College of Engineering & Applied Science, and is a Senior Staff Software Engineer at Zoox. Dr. Heckman earned his BS in Mechanical Engineering from UC Berkeley in 2008 and his PhD in Theoretical and Applied Mechanics from Cornell University in 2012, where he was an NSF Graduate Research Fellow. Dr. Heckman had postdoctoral appointments at the Naval Research Laboratory in Washington, DC as an NRC Research Associate, and at CU Boulder as a Research Scientist, before joining the faculty there in 2016. Dr. Heckman was previously a Visiting Academic with Amazon Scout, where he is addressing last-mile delivery by developing autonomous robots on sidewalks, and is an NSF CAREER Award recipient.