# SceneSense: Diffusion Models for 3D Occupancy Synthesis from Partial Observation

Alec Reed Brendan Crowe Doncey Albin Lorin Achey Bradley Hayes Christoffer Heckman

***Abstract*— When exploring new areas, robotic systems generally exclusively plan and execute controls over geometry that has been directly measured. When entering space that was previously obstructed from view such as turning corners in hallways or entering new rooms, robots often pause to plan over the newly observed space. To address this we present SceneScene, a real-time 3D diffusion model for synthesizing 3D occupancy information from partial observations that effectively predicts these occluded or out of view geometries for use in future planning and control frameworks. SceneSense uses a running occupancy map and a single RGB-D camera to generate predicted geometry around the platform at runtime, even when the geometry is occluded or out of view. Our architecture ensures that SceneSense never overwrites observed free or occupied space. By preserving the integrity of the observed map, SceneSense mitigates the risk of corrupting the observed space with generative predictions. While SceneSense is shown to operate well using a single RGB-D camera, the framework is flexible enough to extend to additional modalities. SceneSense operates as part of any system that generates a running occupancy map 'out of the box', removing conditioning from the framework. Alternatively, for maximum performance in new modalities, the perception backbone can be replaced and the model retrained for inference in new applications. Unlike existing models that necessitate multiple views and offline scene synthesis, or are focused on filling gaps in observed data, our findings demonstrate that SceneSense is an effective approach to estimating unobserved local occupancy information at runtime. Local occupancy predictions from SceneSense are shown to better represent the ground truth occupancy distribution during the test exploration trajectories than the running occupancy map. Finally, we analyze example predictions and show that SceneSense provides reasonable, accurate, and useful predictions.**

## I. Introduction

Humans rely extensively on 'common sense' inferences to engage successfully with the world, while robots are limited to making decisions over directly measured data, such as those captured by cameras or lidar. Humans' natural capability to logically extend geometry or terrain in familiar environments such as homes or offices allows for planning beyond direct observation. In this work, we propose a solution addressing this important technical gap, to expand the scope of situations where autonomy can succeed.

Recent advances in AI systems that generate open-ended representations, known as generative AI, give us the building blocks to develop a generative model for predicting out of view or occluded geometry. Previous attempts at generating "extended terrain" borrowing from point cloud completion (PCC) methods [1] struggle to generalize to new environments. Existing methods for generating out-of-view geometry such as semantic scene completion (SSC) [2], [3], [4] and more recently scene synthesis based approaches [5], [6] are promising. However, SSC is limited as it is a completion or hole-filling method applied only to the frustum of the sensor, rather than a truly generative approach that allows for full, 360° occupancy prediction. For their part, synthesis-based approaches require many views for scene synthesis and are not usable as an online method due to slow inference speed.

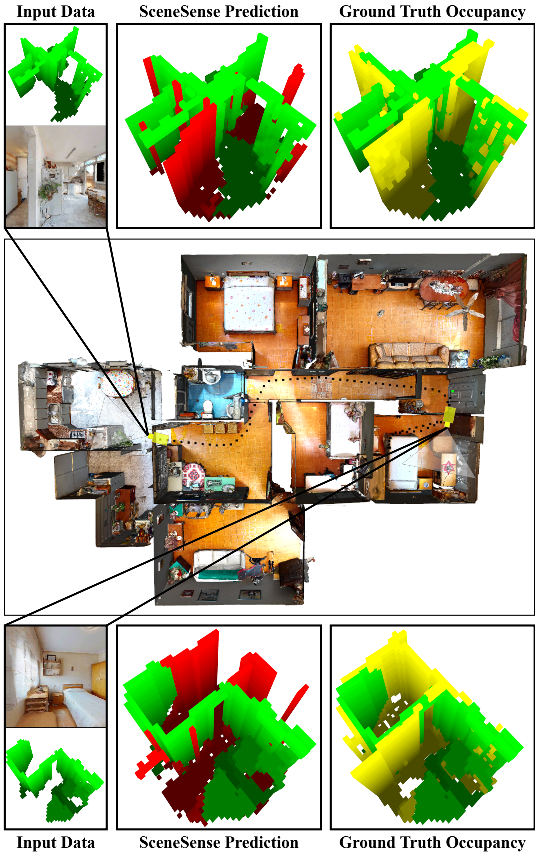


Fig. 1: Test house 2 where the robot exploration trajectory is shown via the black points, and the starting point is shown as green. Two SceneSense generations are shown. From left to right (1) Inputs are on the left where green voxels are the local occupancy information as well as the current camera view from the robot. (2) SceneSense occupancy prediction is shown where occupancy information is shown in green and new predicted occupancy is red. (3) The running occupancy information is again shown in green and the ground truth full local occupancy data is shown in yellow.

*This work was supported by NSF Award #1932189.

All authors are with the Intelligent Robotics Laboratory, Department of Computer Science, at the University of Colorado Boulder, firstname.lastname@colorado.edu

Motivation for development of these generative models can be found in the results of the DARPA subterranean (SubT) challenge [7]. The SubT challenge tasked robot teams with the goal of locating human artifacts after being released into various unknown environments. These search and rescue missions provide a challenging and impactful venue for the deployment of autonomous systems. While teams were fairly successful in locating artifacts, the search took place with a long tail of artifact discovery over an hour. Systems over-searched areas to ensure maximum volumetric frontier gain and frequently paused when there was an influx of new information (such as turning corners into hallways or entering new rooms) [8]. It is our hypothesis that platform exploration speeds can be accelerated using generative models for occupancy prediction.

In this work we leverage recent advances in generative AI as well as practically available robotics data to generate a predicted occupancy grid around a robotic platform. We show that even with a single RGB-D camera and limited training data we can develop an occupancy prediction model that effectively infers the existence of geometry that is out-of-view or occluded. During training, Gaussian noise controlled by a noise scheduler [9] is added to ground truth occupancy data to generate noisy occupancy grids. We simultaneously train a U-net and perception backbone to reduce noise in the occupancy grid, conditioned by an RGB-D image. At inference time, features are extracted from the input images using the trained PointNet++ model [10] and used as conditioning during the reverse diffusion process. Additionally, we take advantage of the occupancy map constructed during exploration to perform *occupancy inpainting*, increasing the fidelity of our results. Critically, using an inpainting approach ensures that the predicted occupancy grid around the robotic platform will never be modified in areas of the scene that have been directly observed to be occupied or free. Finally, we evaluate our framework in home environments from the HM3D dataset [11] against a running occupancy map. Our results show that our proposed approach (*SceneSense*) enhances the local occupancy predictions around the platform. The primary contributions of this work are as follows:

1) A generative framework for estimating out of view or occluded occupancy around the robotic platform.
2) A diffusion inference method we call *occupancy inpainting* that both enhances the predictions of the generative framework and ensures predictions will never overwrite observed free or occupied space.
3) An extensive ablation study outlining the performance trade offs of various tunable parameters that are configurable at runtime.

## II. Related Works

### A. Semantic Scene Completion (SSC)

SSC seeks to generate a dense semantically labeled scene in a target area given some sparse scene representation in that area. Generally the provided information is quite sparse and requires the SSC models to fill in large gaps due to occlusions from the viewpoint. SSC methods are often designed specifically for outdoors [2] or indoor applications [12]. While outdoor SSC implementations focus on SSC using a 3D lidar, indoor methods use aimed sensors such as a RGB-D camera. Due to the shape of this input data outdoor models focus on prediction and labeling all voxels in a grid around the platform, while indoor models generally focus on performing SSC in the frustum of the sensor. Predicting correct geometry and semantic labels is a challenging task. SSC methods have been noted to suffer from poor performance given the challenging nature of the problem [13]. Indoor methods perform around 40% mIoU [13] when "filling in" data in the frustum of the sensor, but are not intended to be generative models that can expand a partially observed scene. Our method expands the role of these models to not only fill in occluded information in the sensor's field of view but to also generate a prediction of what geometry may look like around the platform.

### B. Generative 3D Scene Synthesis

View synthesis is a field of study that seeks to construct a 3D scene from multiple camera views. This is primarily accomplished using a deep learning framework called Neural Radiance Fields (NeRFs) [5]. Original works in this space synthesized 3D views of objects from multiple camera views [14] , while recent works have synthesized full indoor scenes [15]. Current NeRF implementations are slow at inference time, often taking on the order of minutes to render a scene [5]. This characteristic makes them unsuited to real-time scene rendering. In addition, NeRFs require multiple views of the environment to generate reliable volumetric scenes, which may not be available when operating in real-time.

Recently, diffusion models have been explored as a means for synthesising scenes [6]. Initial studies show these models outperform traditional models in scene synthesis and introduce popular generative metrics to the scene synthesis research space. While these implementations are not directly usable in a robotics context, the ideas and evaluation metrics for 3D scene synthesis are applicable to our problem space.

### C. 3D Diffusion and Diffusion in Robotics

Diffusion models [9], [16] have had great success as deep generative models. Diffusion models have generated impressive results across diverse modalities, such as image [17] video [18], audio [19] and natural language [20]. A number of surveys have been published in recent years providing further details on various implementations [4], [21]. Research on diffusion in 3D is limited and generally applied to single object generation [22]. However recent works have begun to explore application of diffusion models for scene generation. Notable work in this space include LegoNet [23] which applies diffusion models to rearrange objects in a 3D scene and DiffuScene [6] which supports unconditional or prompted diffusion of 3D scenes. While these diffusion methods are effective at their task they do not directly translate to robotic applications due to inference time requirements and the required conditioning data.

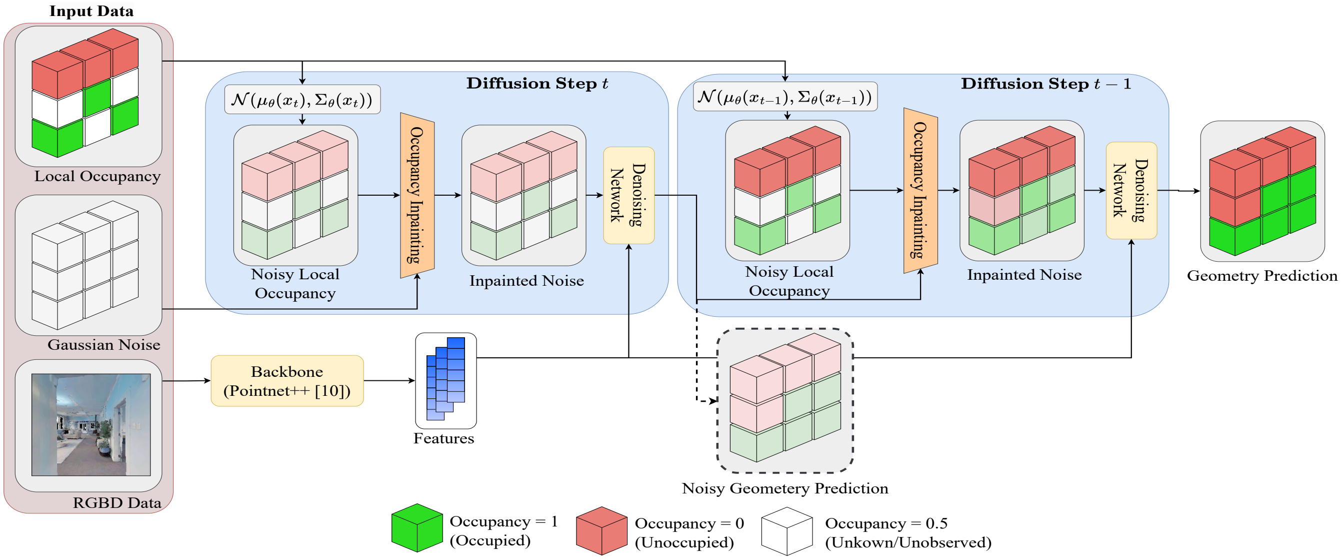


Fig. 2: **Reverse Diffusion Process**: The reverse diffusion process takes the local occupancy information, the current sensor measurements (RGB-D image in this case) and the Gaussian noise of the area to be diffused over. Noise commensurate with the current diffusion step is added to the local occupancy information, which includes occupied (green) and observed unoccupied (red) data. The result is inpainted into the noisy local occupancy prediction as discussed in section IV. The inpainted noise data and the feature vectors generated by the perception backbone are provided to the denoising network which generates a new noisy geometry prediction at $t-1$. This processes is repeated as the starting noise $x_T$ is iteratively denoised to $x_0$ which is the final geometry prediction from the framework.

**Diffusion for Robotic Applications**. The success of diffusion models have inspired researchers to begin to apply them in the robotic domain. While the Markovian nature of diffusion models can be a bottleneck for systems requiring real-time inference diffusion models have been successfully applied to real-time robotics problems such as planning problems [24], [25] and perception [26], [27]. These generative models are both increasing the effectiveness of current methods in traditional robotics problems as well as providing new research areas to tackle.

## III. Preliminaries and Problem Definition

### A. Problem Definition: Dense Occupancy Prediction

The objective of dense occupancy prediction is to predict the occupancy from $[0,1]$ where 0 is unoccupied and 1 is occupied for every voxel $v$ in a target region $x$ where $v \in \mathbb{R}^{\mathbf{z}\times\mathbf{x}\times\mathbf{y}}$.

### B. Forward Diffusion

$x_0$ is defined as a clean occupancy grid where the distribution of $x_0$ can be defined as $q(x_0)$. 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 $t \in [1, T]$, $\beta_t$ 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 $\epsilon$ is the noise used to corrupt $x_t$.

### C. Reverse Diffusion

Reverse diffusion is a Markov chain of learned Gaussian transitions $p_\theta(x_{t-1}|x_t)$ which is parameterized by a learnable network $\theta$:

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

where $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are the predicted mean and covariance respectively of the Gaussian $x_{t-1}$. 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 on of the reverse diffusion process is simply the product of the applied learned Gaussian transitions $p_\theta(x_{t-1}|x_t)$:

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

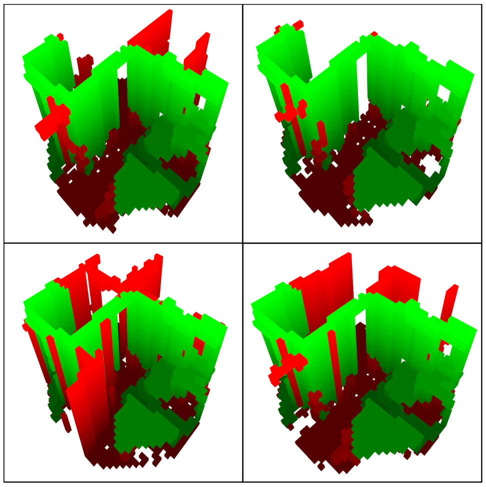


Fig. 3: Various SceneSense predictions from equivalent input data where green is the running occupancy map and red is the SceneSense predicted occupancy. Given the limited input information the diffusion framework can generate multiple reasonable predictions from the same input conditioning.

### D. *Conditional Diffusion*

The conditional diffusion model extends the diffusion process to guide the diffusion by some conditioning $y$. In particular we use a class of conditional diffusion models called classifier-free diffusion [28]. During training the diffusion model $f_\theta(x_t, y, t)$ is trained to predict $x_0$ from $x_t$ under the guidance of condition $y$. During training conditioning $y$ is replaced with a null label $\emptyset$ with a fixed probability. At inference time $x_0$ is reconstructed from $x_T$ with guidance from the conditioning $y$. During sampling at inference time the output of the model is "pushed" toward the conditional model result $f_\theta(x_t|y)$ and away from the unconditioned result $f_\theta(x_t|\emptyset)$ as follows:

$$\hat{f}_\theta(x_t|y) = f_\theta(x_t|\emptyset) + s \cdot (f_\theta(x_t|y) - f_\theta(x_t|\emptyset)), \tag{6}$$

where $s$ is the guidance scale defined as $s \in \mathbb{R}_{\geq 0}$. $s$ is a configurable parameter at runtime that allows for the user to configure how closely the model should adhere to the provided conditioning.

## IV. Method

### A. *Architecture*

**Denoising Network**. The denoising network in our method is inspired by the popular image generation diffusion network Stable Diffusion [17]. It is a U-net constructed from the HuggingFace Diffusers library of blocks [29] and consists of Resnet [30] downsampling/upsampling blocks with cross-attention as well as regular ResNet downsampling/upsampling blocks. The conditioning features generated by Pointnet++ are mapped to the intermediate layers of the U-net via the cross attention layers of the transformer blocks as discussed by Stable Diffusion [17].

**Feature Extraction and Conditioning**. As discussed in section III classifier-free diffusion models are conditioned on a set of guidance data $y$. Our model uses the Pointnet++ [10] backbone to generate a $N \times F$ feature matrix from a given RGB-D image, where $F$ is the number of desired Pointnet++ features per point $n \in N$. As is common with other vision based pipelines [31] the feature generation backbone can be replaced with other models to increase performance or add/remove different types of perception modalities. Further, the conditioning of the diffusion model can be any modality encoding into feature vectors. Conditioning could include standard robotic sensors such as camera, lidar or radar, but could also be extended to include informational modalities such as human text input or sketches of the scene [17].

**Occupancy Mapping**. Occupancy mapping allows for 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 [32] 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 be maintaining 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.

### B. *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 $f_\theta$ to predict the noise applied to $x_t$ given the associated RGB-D conditioning $y$.

**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 $\alpha_t$ which is configured by a linear noise scheduler [9].

**Loss Function**. The network $f_\theta$ 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 [33], [17], [27], [26].

### C. *Inference*

**Sampling Process** The trained noise prediction network $f_\theta$ 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 $f_\theta$ and the associated RGB-D features $y$ to compute $x_{t-1}$ The RGB-D features $y$ are applied as conditioning during the process with the cross-attention mechanism [34].

**Occupancy Inpainting**. Our method of occupancy inpainting ensures observed space is never overwritten with SceneSense predictions. Additionally occupancy inpainting enhances the predictions from the models seen in fig. 6 (c). Inspired by image inpainting methods seen in image diffusion [17] and guided image synthesis methods [35], occupancy inpainting continuously applies the known occupancy information to the diffusion target during inference. To perform occupancy inpainting we sub-select a section of our occupancy map that we will perform diffusion over. In that area we take the occupied map $M_o$ and unoccupied map $M_u$ to generate occupied and unoccupied masks, $Ma_o$ and $Ma_u$ respectively. These masks will be applied to the target the diffusion to only allow occupied and unoccupied voxels to be modified during each masking step. Finally we add noise commensurate with the current diffusion step and update the diffusion target with the noisy occupied and unoccupied voxel predictions. This process is repeated for each inference step and can be seen in fig. 2. This method both increases the fidelity of the scene predictions and ensures the diffusion model does not predict or modify geometry in space that has already been observed.

**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 fig. 3 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. For simplicity we simply generate one prediction at each pose, but additional heuristics or voting schemes could be added to the system to score multiple outputs and select preferable predictions.

## V. Experiments

We use the Habitat lab simulation platform [36] and the Habitat-Matterport 3D research dataset (HM3D) Dataset [11] to generate training and test data. We operate a simulated platform with a 256×256 RGB-D camera through 12 different house environment to generate full occupancy grids with voxel resolution of $0.1m$ of the homes as well as ≈9000 poses and associated RGB-D camera views to be used as conditioning. We split the dataset into a training and test set by house number. Houses 3–12 are used in the training set and houses 1 and 2 as shown in fig. 4 and fig. 1 respectively are used as the test set. For testing arbitrary trajectories are taken to navigate through each home, and SceneSense is used to predict the local occupancy information at each timestep.

**Implementation**. The diffusion model is trained using randomly shuffled pairs of conditioning $y$ and ground truth occupancy grids $x$, where various houses may be mixed in a batch. We use Chameleon cloud computing resources [37] to train our model on one A100 with a batch size of 16 for 250 epochs or 119,250 training steps. We use a cosine learning rate scheduler with a 500 step warmup from $10^{-6}$ to $10^{-4}$. We set dropout to 0.2 where the conditioning $y$ is set to $\emptyset$. The noise scheduler for diffusion is set to 1000 noise steps. At inference time we evaluate our dataset using an RTX 4090 GPU for acceleration. On average we measure a diffusion step for our model to be 0.0633 seconds.

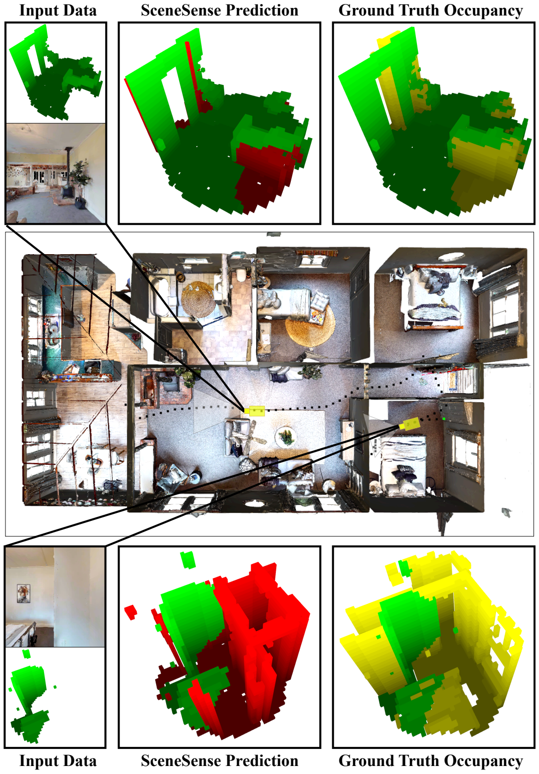
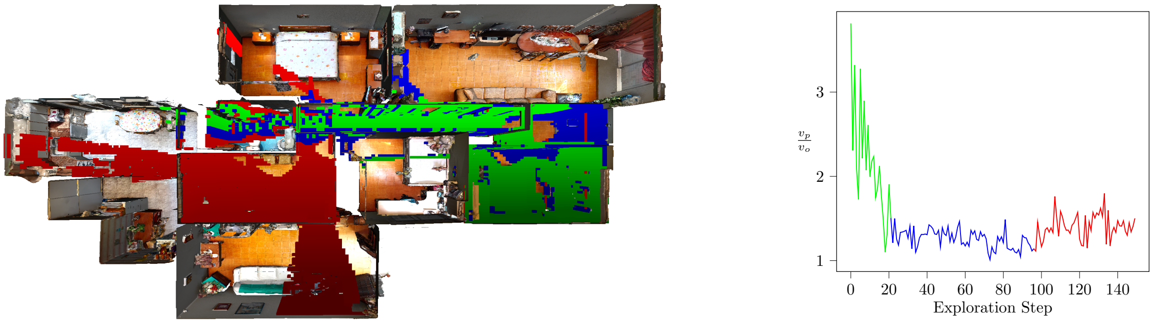


Fig. 4: Test house 1 where the robot explore trajectory is shown via the black points, and the starting point is shown as green. Two SceneSense generations are shown. From left to right (1) Inputs are on the left where green voxels are the local occupancy information as well as the current camera view from the bot. (2) SceneSense occupancy prediction is shown where occupancy information is shown in green and new predicted occupancy is red. (3) The running occupancy information is again shown in green and the ground truth full local occupancy data is shown in yellow.

**Baselines**. To our knowledge this is the first architecture to apply a generative method to predict 3D occupancy around a platform from a single aimed sensor. This makes direct comparisons for performance difficult to evaluate. As such, the best evaluation of our method is an evaluation against the running octomap (BL). Improvement upon this baseline shows that occupancy predictions from SceneSense better represents the ground truth occupancy information at a given pose than the running occupancy grid. The code and inference dataset will be released upon publication https://arpg.github.io/scenesense/.

**Evaluation**. Following similar generative scene synthesis approaches [6], [38] we employ the Fréchet inception distance (FID) ] [39] and the Kernel inception distance [40] (KID ×1000) to evaluate the generated local occupancy grids using the clean-fid library [41]. Generating good metrics to evaluate generative frameworks is a difficult task [42].



(a)

(b) **Conditioning**

Fig. 5: Calculated predicted voxels $v_p$ over occupied voxels $v_o$ ($\frac{v_p}{v_o}$) over the house 2 exploration. (a) Superimposes the running occupancy map over the house mesh where the colors of the occupancy map show how many steps have ran to that point. Green voxels are the running occupancy map from step 0 to step 20, blue are step 0 to step 95, and red are step 0 to 150. These colors correspond with the plot line colors in (b). (b) Shows the $\frac{v_p}{v_o}$ as the robot explores the space. $\frac{v_p}{v_o}$ starts high at time step 0, when the occupancy map is sparse, and quickly drops over the green exploration where more of the local scene is observed. $\frac{v_p}{v_o}$ stays relatively low as the vehicle completes the exploration of the green room, navigates back to the start point and traverses the hallway. $\frac{v_p}{v_o}$ increases slightly as the robot traverses previously unobserved space (red), which requests more predicted voxels as less of the scene has been observed.

FID and KID have become the standard metric for many generative methods due to their ability to score both accuracy of predicted results, as well as 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 show that they are an effective measure of the success of a generative framework like SceneSense.

TABLE I: Quantitative comparisons of local occupancy synthesis from two test home environments from the HM3D [11] dataset.

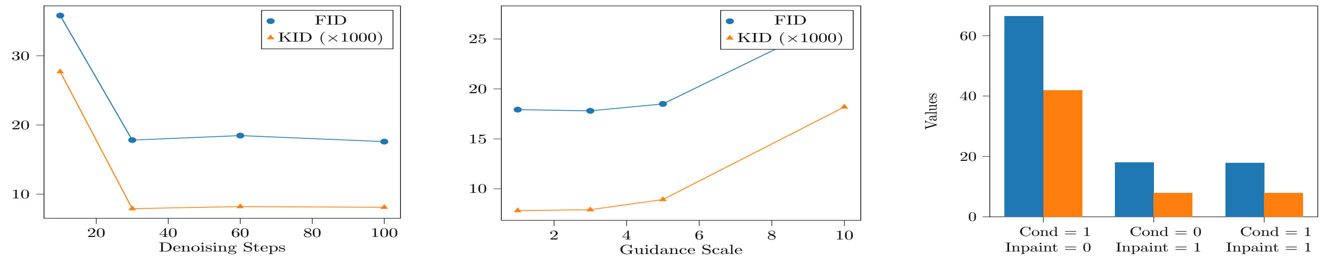
| | House 1 | | House 2 | |
|---|---|---|---|---|
| Method | FID ↓ | KID ↓ | FID ↓ | KID ↓ |
| BL | 26.18 | 18.91 | 22.55 | 14.06 |
| SS | 17.81 | 7.93 | 20.94 | 6.93 |

## VI. Results

Quantitative results for the test set are shown in table I while qualitative results showing example diffused occupancy grids can be seen in fig. 4 and fig. 1. SceneSense achieved substantial reductions in both KID and FID when compared to the baseline running occupancy method. Additionally, the qualitative results show reasonable estimates of potential geometry around the platform given limited input information.

**Quantitative Discussion**. The results presented in table I favor SceneSense when compared to the local running occupancy information in both scenes tested. In house 1 substantial reduction in FID and KID, 31% and 58% respectively, were reported. While the reduction in FID was lower in house 2 (7%) the reduction in KID was similar to that measured in house 1, 50%. KID is known to be less sensitive to outliers and is considered by some to be an advantageous metric for evaluating generative frameworks when compared to the FID [40]. As shown in fig. 1 house 2 is a much different layout than house 1, with many small hallways, rooms and corners. It is likely that house 1 is more similar to those captured in the training set than house 2, leading to an increase of erroneous predictions in the house 2 scene. These predictions result in a larger FID score while the KID remain low due to the native robustness to these skewed distributions.

Further we can look at example predictions for qualitative evaluation and to examine FID and KID as evaluation metrics compared to traditional metrics such as IoU. Take for example the upper prediction in fig. 1 where the platform is entering the kitchen. SceneSense adds useful information to the problem, predicting the existence of floor as well as a wall that would obstruct motion to the right of the robot. However the wall is 0.1 - 0.2m off the actual existing wall. The result of this incorrect locating results in a worse IoU (0.52) than the local occupancy information (0.61). These problems in quantitative evaluation are only exacerbated when there is more geometry to estimate such as in the bottom observation of fig. 4. Generative models incur a large penalty in the IoU metric for guessing at geometry, even when given very little context for the prediction. Predictions increase the total union of the space, however a missed prediction does not increase the intersection of the space. In the case of the top prediction in 1 even if the prediction would be useful in planning, since the wall is mislocated the metric reports a much worse IoU than the running occupancy. Additionally in cases where little information is known, such as in the bottom prediction of 4, an occupancy prediction of what the geometry may look like is heavily penalized since the metric does not evaluate if a prediction distribution is reasonable. However the FID and KID metrics evaluate both the accuracy of predictions as well as the distribution of the predictions, allowing for generative frameworks that attempt to generate challenging results to be fairly scored. FID and KID are already widely adopted metrics in generative fields such as image generation [43], [17] and scene synthesis [6], [38] for these reason and our results support their use in this



(a) **Denoising steps**

(b) **Guidance scale** $s$

(c) **Conditioning**

Fig. 6: **SceneSense ablation experiments**: All ablation experiments were run on test house 1 using the same trained diffusion network. For all experiments conditioning and inpainting are enabled, $s$ is set to 3 and 30 denoising steps are used unless these values are being ablated. (a) Figure (a) ablates various denoising step values. (b) Figure (b) ablates various guidance scale $s$ values as defined in eq. (6). (c) Figure (c) ablates enabling conditioning and inpainting for the network. A 1 indicates the value is enabled and 0 indicates it is disabled.

context.

**SceneSense Accuracy Over Time**. As discussed in section IV SceneSense only predicts occupancy in areas that have not been directly measured to be occupied or unoccupied. This means that as exploration of the space approaches 100% SceneSense will have no unobserved space to predict over. This can be measured as $\frac{v_p}{v_o}$ where $v_p$ are the predicted voxels from SceneSense and $v_o$ are the local occupied voxels. Given 100% exploration this metric will approach 1 where all predicted voxels are simply the local occupied voxels. This reduction in prediction space is shown in fig. 5. Initially $\frac{v_p}{v_o}$ is quite high, since very little of the scene has been observed but quickly drops during exploration. Spikes in $\frac{v_p}{v_o}$ are seen when the platform moves to new areas that have occlusions such as hallways or when entering new rooms. These metrics support the assertion that SceneSense respects measured space and only generates geometry where no measurement has been taken.

### A. Ablations

**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 [27], [26]. As shown in fig. 6 (a) 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.

**Conditioning and Guidance Scale Discussion**. The guidance scale $s$ as defined in eq. (6) 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 area in front of the to geometry 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.

## VII. Conclusions and Future Work

In this paper we present SceneSense; a diffusion-based approach for generative local occupancy prediction. SceneSense is shown to enhance local occupancy information quantitatively using standard metrics for generative AI, as well as qualitatively by providing frames generated by the framework. While past methods have focused on filling holes in observed information, or synthesising scenes offline from multiple camera views, SceneSense can quickly infer scene information 360° around the platform. Future work for SceneSense includes integration with a planning and control architectures for scene exploration and testing of additional conditioning modalities such as language to further enhance scene predictions.

## References

[1] Alec Reed and Christoffer Heckman. Looking around corners: Generative methods in terrain extension, 2023.

[2] Ran Cheng, Christopher Agia, Yuan Ren, Xinhai Li, and Liu Bingbing. S3cnet: A sparse semantic scene completion network for lidar point clouds. In Jens Kober, Fabio Ramos, and Claire Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 2148–2161. PMLR, 16–18 Nov 2021.

[3] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image, 2016.

[4] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications, 2023.

[5] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022.

[6] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Scene graph denoising diffusion probabilistic model for generative indoor scene synthesis, 2023.

[7] Timothy H Chung, Viktor Orekhov, and Angela Maio. Into the robotic depths: Analysis and insights from the darpa subterranean challenge. *Annual Review of Control, Robotics, and Autonomous Systems*, 6:477–502, 2023.

[8] Harel Biggie, Eugene R. Rush, Danny G. Riley, Shakeeb Ahmad, Michael T. Ohradzansky, Kyle Harlow, Michael J. Miles, Daniel Torres, Steve McGuire, Eric W. Frew, Christoffer Heckman, and J. Sean Humbert. Flexible supervised autonomy for exploration in subterranean environments, 2023.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[10] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017.

[11] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[12] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 3d sketch-aware semantic scene completion via semi-supervised structure prior, 2020.

[13] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. 3d semantic scene completion: a survey, 2021.

[14] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020.

[15] Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views, 2022.

[16] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[18] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos, 2022.

[19] Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. Prodiff: Progressive fast diffusion model for high-quality text-to-speech, 2022.

[20] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2023.

[21] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey, 2023.

[22] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation, 2022.

[23] Sihao Yu, Fei Sun, Jiafeng Guo, Ruqing Zhang, and Xueqi Cheng. Legonet: A fast and exact unlearning architecture, 2022.

[24] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis, 2022.

[25] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

[26] Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo. Ddp: Diffusion model for dense visual prediction. *arXiv e-prints*, pages arXiv–2303, 2023.

[27] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*, 2022.

[28] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[29] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[31] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[32] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. Software available at https://octomap.github.io.

[33] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR, 17–23 Jul 2022.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[35] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

[36] Xavi Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Ruslan Partsey, Jimmy Yang, Ruta Desai, Alexander William Clegg, Michal Hlavac, Tiffany Min, Theo Gervet, Vladimir Vondrus, Vincent-Pierre Berges, John Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023.

[37] Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S. Gunawi, Cody Hammock, Joe Mambretti, Alexander Barnes, François Halbach, Alex Rocha, and Joe Stubbs. Lessons learned from the chameleon testbed. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20)*. USENIX Association, July 2020.

[38] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. In *2021 International Conference on 3D Vision (3DV)*, pages 106–115. IEEE, 2021.

[39] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[40] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

[41] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, 2022.

[42] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models, 2020.

[43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.