

ForesightNav: Learning Scene Imagination for Efficient Exploration

Hardik Shah
ETH Zurich

Jiaxu Xing
University of Zurich

Marc Pollefeys
ETH Zurich, Microsoft

Nico Messikommer
University of Zurich

Boyang Sun
ETH Zurich

Davide Scaramuzza
University of Zurich

Abstract

Understanding how humans leverage prior knowledge to navigate unseen environments while making exploratory decisions is essential for developing autonomous robots with similar abilities. In this work, we propose **ForesightNav**, a novel exploration strategy inspired by human imagination and reasoning. Our approach equips robotic agents with the capability to predict contextual information, such as occupancy and semantic details, for unexplored regions. These predictions enable the robot to efficiently select meaningful long-term navigation goals, significantly enhancing exploration in unseen environments. We validate our imagination-based approach using the Structured3D dataset, demonstrating accurate occupancy prediction and superior performance in anticipating unseen scene geometry. Our experiments show that the imagination module improves exploration efficiency in unseen environments, achieving a 100% completion rate for PointNav and an SPL of 67% for ObjectNav on the Structured3D Validation split. These contributions demonstrate the power of imagination-driven reasoning for autonomous systems to enhance generalizable and efficient exploration. Project code is available at [uzh-rpg/foresight-nav](https://github.com/uzh-rpg/foresight-nav).

1. Introduction

Human navigation in novel environments is a complex process guided by a combination of explicit maps and accumulated geometric and semantic information. This prior information allows us to infer spatial layouts and anticipate the presence and location of objects based on context and experience. For example, we intuitively expect to find toilets and showers in bathrooms or look for a pot in the kitchen, even without seeing them first. Given only a partial view, humans use their *imagination* to reason about unexplored

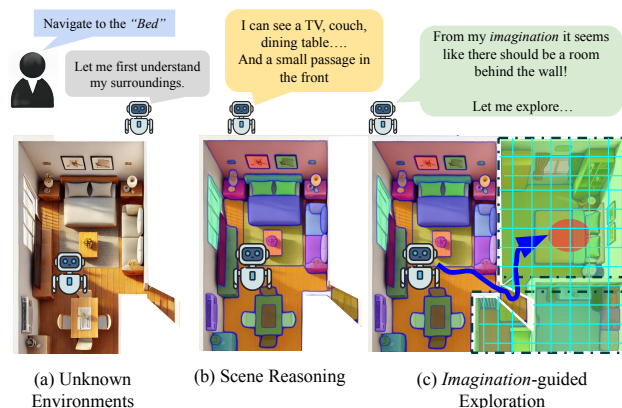


Figure 1. ForesightNav proposes *Imagination* aided exploration in unknown environments. The past observations by the agent are used to reason about the scene and extract geometric and semantic information. The agent uses this information to *imagine* the structure of the unobserved regions, enhancing its ability to explore efficiently.

areas, predicting unseen regions based on prior knowledge to guide navigation and exploration decisions [11]. Extending this imagination capability to autonomous robots would enhance their ability to understand and interpret complex environments. Such improved understanding directly translates to more efficient autonomous robotic navigation. Consequently, incorporating imagination-based capabilities into robotic systems is promising for real-world applications such as home service, search and rescue, delivery, and industrial inspections [25, 33].

However, accurately predicting unseen regions from limited or partial observations remains challenging, as it requires robust generalization to novel environments. Moreover, developing suitable representations and training data for imagination networks that effectively include geometric structure and semantic information is inherently com-

plex. Existing navigation approaches have progressed by integrating scene information in an end-to-end fashion, often via reinforcement learning (RL). However, these end-to-end pipelines frequently fail to generalize and demand extensive retraining for each new scenario [3, 6, 37]. In contrast, advanced Vision-Language Models (VLMs) offer zero-shot solutions that circumvent task-specific training, though they may struggle with navigation precision due to their limit in understanding spatial information [26, 30, 38].

Very few prior works [12, 21, 40] have focused on imagining explicit map representation from partial observation, like we humans naturally do and have limitations in their scalability and evaluation. To address this, we propose a novel exploration strategy for autonomous navigation that equips agents with a learned imagination capability, as illustrated in Figure 1. Our method models imagination as a neural network that leverages current and past observations to predict unobserved regions of the environment, guiding more informed and goal-directed exploration. Our approach comprises two main components: *predicting scene geometry* from partially observed occupancy maps and enhancing this prediction with *semantic understanding* using visual language features. Inspired by human intuition, our approach forms beliefs about unobserved areas based on semantic relationships and spatial understanding, facilitating efficient and informed exploration. The Imagination Module is compatible with predicting both the unseen floor plan geometry and semantic visual-language (CLIP) features [28] from partially observed information. This unified prediction allows the agent to efficiently select long-term navigation goals using limited partial observation. Our results demonstrate superior performance in point goal navigation in unseen environments. The imagination strategy significantly improves navigation efficiency by predicting unseen scene geometry from partial occupancy information. We outperform SOTA approaches in the Object Goal Navigation task on the Structured3D dataset that requires both semantic and geometric analysis. To summarize, our contributions are:

1. We propose **ForesightNav**, a novel approach for exploration based on **Imagination** of the scene representation from past observations, where an agent goes beyond reactive strategies and actively reasons about the unseen environment.
2. We propose a unique method for the training of occupancy and semantic prediction of indoor scenes by **generating partially observed occupancy masks** involving the simulation of an agent in an unseen 2D grid world of a scene to reach uniformly distributed goal waypoints.
3. We introduce a new **closed-loop evaluation benchmark** for the task of Object Goal Navigation in a 2D grid world using the Structured3D dataset, with 3500 indoor scenes. Our benchmark on Structured3D enables a more

comprehensive evaluation than existing benchmarks, improving the assessment of method generalization

2. Related Work

2.1. Object Goal Navigation

Object Goal Navigation (ObjectNav) focuses on navigating to a specified object in a novel environment, utilizing semantic priors to improve navigation efficiency. Traditional approaches leverage reinforcement learning [5, 8, 16, 35], learning from demonstration [29], or creating semantic top-down maps [6, 39] to assist in waypoint planning. These methods, however, are limited to the specific object categories they were trained on and often depend on simulated data, hindering real-world application.

To overcome these limitations, zero-shot ObjectNav methods adopt frontier-based exploration. Classical methods select exploration frontiers [34] based on potential information gain [24]. Recent methods like CLIP on Wheels (CoW) [13] use CLIP features or an open-vocabulary object detector to identify target objects during frontier exploration. Other works, such as LGX [9], ESC [42], and StructNav [7], incorporate large language models (LLMs) to score frontiers based on object detections. However, these approaches require converting visual features to text, introducing computational bottlenecks and often requiring remote server access. In contrast, our approach leverages a language-driven semantic segmentation encoder, LSeg [22] that operates directly on RGB observations to generate semantic embeddings in the CLIP [27] feature space. This method eliminates the need for text generation from visual data and can run efficiently on consumer hardware, facilitating real-time zero-shot navigation. Recent advancements include VLFM [36], which constructs occupancy maps from depth observations and uses a vision-language model to select exploration frontiers based on semantic values extracted from RGB images. This method facilitates open-vocabulary navigation, unlike earlier approaches that are restricted to a fixed set of object categories. Our approach is different from these methods in integrating the imaginative capability of humans into robotic exploration. The Imagination Module predicts unseen parts of the environment using current and past observations, enhanced with semantic understanding via *GeoSem Maps* - a vision-language grounded scene representation that consolidates geometry and semantics, providing a more robust and adaptable solution for object goal navigation in diverse environments.

2.2. Scene Representation

Effective scene representation is crucial for robotic navigation. Various methods have been proposed, such as [14], which uses segments from images as nodes in a topological map. [20] and [17] employ multi-modal scene rep-

representations combining CLIP [27] and DINO [4] embeddings to enhance 3D mapping and semantic understanding. These methods, however, require pre-exploration and mapping of the environment, limiting their applicability in real-time navigation tasks. In [15], it is shown that modular approaches that use an intermediate scene representation outperform end-to-end approaches and transfer the best from sim to real, which further reinforces our decision to use *GeoSem Maps* as an intermediate representation. Apart from the visual language features, spatial information such as floor plans are often used as well in the navigation task. One method extends maps by 2 meters in all directions using generative models to address environmental uncertainty [21], but it falls short of producing complete floor plans, limiting scalability. Another approach predicts top-down semantic maps trained with randomly masked inputs [40], yet such masking does not mirror real-world observations. We instead simulate agents navigating environments to generate realistic partial views. A different method infers unseen walls as 2D line segments from 360° LIDAR data [12], but treats floor plan prediction as a vision task with metrics misaligned with actual navigation. [32] also treats the prediction of Open-Vocabulary 3D Occupancy BEV Maps in outdoor scenes as a computer vision task. Our method evaluates imagination by simulating navigation performance of the agent in novel environments, offering a more practical and task-driven benchmark.

3. Methodology

Problem Formulation

In this work, we address the Object Goal Navigation (ObjectNav) task [3], where an agent is tasked with locating an instance of a specified object category (e.g., ‘bed’, ‘chair’) within a previously unseen environment. The agent is initialized at a random location and is provided with the target object category as an input goal.

At each time step t , the agent receives visual observations, consisting of first-person RGB and depth images and pose readings, which indicate its position and orientation relative to its starting point. The action space \mathcal{A} includes: MOVE FORWARD (0.25m), TURN LEFT (30°), TURN RIGHT (30°), LOOK UP (30°), LOOK DOWN (30°), and STOP. The objective is for the agent to navigate towards the goal object and take the STOP action when it believes it has reached within 1 meter of the target, with a maximum of 500 steps allowed.

System Overview

Our method, shown in Figure 2, mainly consists of two building blocks: (i) *GeoSem Map*: a scene representation that acts as a memory for the agent for jointly saving geometry and semantics from past observations. In this work, we

use Occupancy and CLIP [27] as the geometry and semantic representations, respectively. (ii) *Imagination module*: the exploration strategy for deducing the long-term navigation goal. As the agent moves through the environment, it records its observations in a GeoSem Map. The imagination module then predicts the complete GeoSem Map of the entire scene based on the current, partially observed one. Using this predicted GeoSem Map, the agent selects a long-term navigation goal to explore the path to the target object category efficiently. The agent subsequently uses the predicted geometry of the scene to navigate to the chosen goal. Once the target object category is within the field-of-view of the agent, it then directly navigates to the object without further exploration.

3.1. GeoSem Map Construction

A GeoSem Map is a Bird’s-Eye-View (BEV) projection of 3D language-aligned voxel features. It acts as a database for the agent to store inferred RGBD observations and to efficiently retrieve the information subsequently during the exploration, similar as done in [19]. Let $\mathcal{X} \subset \mathbb{R}^3$ represent the 3D spatial domain of the environment. Specifically, we define the GeoSem Map as a structured grid-based scene representation $\mathcal{M} \in \mathbb{R}^{H \times W \times (D+1)}$. Here $H \times W$ defines the spatial resolution of the grid, discretizing the environment into uniform cells. D is the dimensionality of the CLIP embedding space. The first D channels at each grid cell store **semantic features**, represented by a function:

$$f_{\text{sem}} : \mathcal{X} \rightarrow \mathbb{R}^D, \quad (1)$$

which maps each spatial location to a D -dimensional CLIP feature vector. The last channel stores **occupancy features**, defined as

$$f_{\text{occ}} : \mathcal{X} \rightarrow \{0, 0.5, 1\}, \quad (2)$$

where $f_{\text{occ}}(x) \in \{0, 0.5, 1\}$, representing free, unknown and occupied cells respectively. Thus, for a 3D point p mapped to the grid cell (i, j) , its representation is

$$\mathcal{M}_{i,j} = [f_{\text{sem}}(p), f_{\text{occ}}(p)] \in \mathbb{R}^{D+1}, \quad (3)$$

where the first D elements of $\mathcal{M}_{i,j}$ encode high-level semantic attributes, and the last element encodes occupancy. We initialize the first D channels of \mathcal{M} with zeros, and the $(D + 1)^{\text{th}}$ channel with 0.5 at the start of each episode, and update it as the episode progresses.

Semantic Mapping f_{sem} : At each timestep t , we compute a dense per-pixel semantic embedding using the LSeg [22] encoder

$$f_{\text{LSeg}} : I_{\text{RGB}}^{(t)} \rightarrow \mathbb{R}^{H_I \times W_I \times D}, \quad (4)$$

where $I_{\text{RGB}}^{(t)}$ is the input image, (H_I, W_I) are its dimensions, and each pixel is mapped to a D -dimensional CLIP-aligned

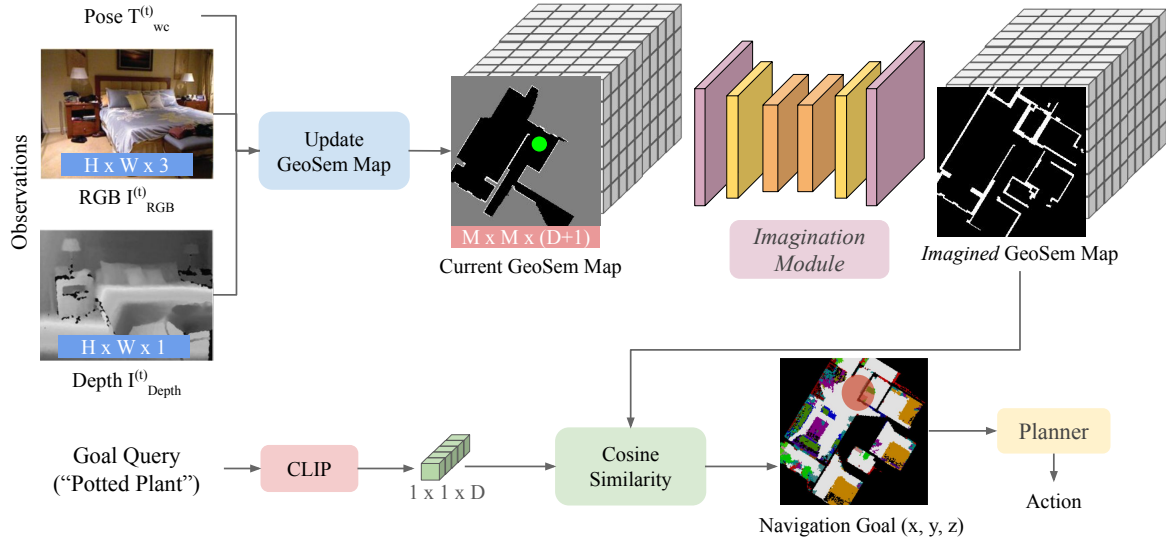


Figure 2. Overview of ForesightNav. At each timestep, we process the RGBD+Pose observations to update our current GeoSem Map, which acts as a database storing scene geometry and semantics. We use the *Imagination Module* to predict the complete GeoSem Map of the scene and use the similarity scores between the Imagined GeoSem Map and the CLIP encoding of the goal query to compute the navigation goal. The planner then deterministically outputs actions to navigate to the goal. We explore only until the agent does not detect the target object.

embedding. Using the depth image $I_{\text{Depth}}^{(t)}$, we project each pixel (u, v) to a local 3D point cloud

$$p^{(t)}(u, v) = d(u, v)K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (5)$$

where $d(u, v)$ is the depth value, and K is the intrinsic camera matrix. Each 3D point inherits the CLIP embedding of its corresponding pixel. To align with our global BEV representation, each point is transformed into the world frame using the camera pose $T_{wc}^{(t)}$,

$$p_w = T_{wc}^{(t)}p^{(t)}. \quad (6)$$

These transformed points are then discretized into the grid map \mathcal{M} with spatial dimensions $H \times W$. If multiple points fall into the same grid cell $\mathcal{M}_{i,j}$, their embeddings are averaged, following the construction of [19].

$$\mathcal{M}_{i,j} = \frac{1}{N_{i,j}} \sum_{k \in \mathcal{S}_{i,j}} f_{\text{sem}}(p_k), \quad (7)$$

where $\mathcal{S}_{i,j}$ is the set of points mapped to grid cell (i, j) , and N forms the density map where $N_{i,j} = |\mathcal{S}_{i,j}|$.

Occupancy Mapping f_{occ} : To compute occupancy, we use the density map N . To ensure relevance to the agent’s navigability, we apply a height-based filter based on the z -coordinate of each point. Only points within a valid height

range $[z_{\min}, z_{\max}]$, determined by the agent type, are considered for occupancy calculation. A cell is then classified as occupied if its density exceeds 10% of the maximum density across all grid cells.

Zero-Shot Object Segmentation: In Figure 3 we show the top-down semantic map and the top-down color map of a scene from the Structured-3D dataset. To generate the top-down semantic map, given K queries, we first encode each query using the CLIP text encoder, resulting in K query vectors. We then compute the dot product of these query vectors with every cell ($H \times W$) of the GeoSem Map \mathcal{M} , yielding K similarity scores for each grid cell. The grid cell is then assigned to the category corresponding to the query with the highest similarity score.

3.2. The Imagination Module

We define the Imagination Module as a learnable function that predicts the ground-truth GeoSem Map of a scene given a partially observed GeoSem Map. This involves predicting the complete occupancy and semantics of the scene. Formally, we define the *Imagination Module* f_{imagine} as

$$f_{\text{imagine}} : \mathbb{R}^{H \times W \times (D+1)} \rightarrow \mathbb{R}^{H \times W \times (D+2)}. \quad (8)$$

Given $\mathcal{J} = f_{\text{imagine}}(\mathcal{M})$, the input \mathcal{M} is a GeoSem Map constructed using a partially observed scene where,

$$\mathcal{M}_{i,j} = [o_{\text{semantic}}, o_{\text{occ}}] \in \mathbb{R}^{D+1}. \quad (9)$$

Here $\mathbf{o}_{\text{semantic}}$, is the first D elements of $\mathcal{M}_{i,j}$ encode already observed CLIP features of the scene. If (i, j) is unobserved, $\mathcal{M}_{ij}[D] = 0$. And $\mathbf{o}_{\text{occ}} \in \{0, 0.5, 1\}$.

The output \mathcal{J} is again a 3D grid representation:

$$\mathcal{J}_{i,j} = [p_{\text{semantic}}, p_{\text{occ}}, p_{\text{interior}}] \in \mathbb{R}^{D+2}, \quad (10)$$

where $\mathbf{p}_{\text{semantic}}$ is the first D elements of $\mathcal{J}_{i,j}$ encode predicted high-level semantic attributes of the scene in the CLIP feature space (including the observed and unobserved cells). $\mathbf{p}_{\text{occ}} \in [0, 1]$, represents the $(D + 1)^{\text{th}}$ element encodes the occupancy probability, and $\mathbf{p}_{\text{interior}} \in [0, 1]$, represents the $(D + 2)^{\text{th}}$ element encodes the probability of a pixel lying inside the scene. The need for p_{interior} is justified later when we define the loss functions.

Generating Training Data: Partially Observed Occupancy Masks A training pair for the f_{imagine} network consists of $(\mathcal{M}, \mathcal{G})$ where $\mathcal{M} \in \mathbb{R}^{H \times W \times (D+1)}$ is the observed GeoSem Map as defined in equation 9 and $\mathcal{G} \in \mathbb{R}^{H \times W \times (D+2)}$ is the ground truth GeoSem-Map along with the ground truth interior mask of the scene, as defined in equation 10 with the true values instead of predicted scores. We can generate multiple instances of \mathcal{M} from \mathcal{G} by masking parts of the occupancy channel $\mathcal{G}_{m_i, m_j, D+1} = 0.5$ and the semantic channels $\mathcal{G}_{m_i, m_j, 1:D} = 0$, where $(m_i, m_j) \in \mathcal{O}$ and \mathcal{O} is a mask. To ensure that we train f_{imagine} on instances that it will encounter during inference for the task of navigation of an agent in an indoor environment, we generate multiple masks \mathcal{O} for every scene in the training split of the dataset by simulating an agent in the scene.

For each training scene, we start by simulating an agent in a 2D grid world - The ground truth occupancy map of each scene, denoted \mathbf{C} . The agent is defined with specific parameters such as ‘field of view (FOV)’, ‘sensor range’, and ‘turn angle’, which mimic real-world sensor and agent configurations. The agent is randomly initialized at an interior point of the scene and is tasked with reaching randomly generated waypoints that are distributed all over the

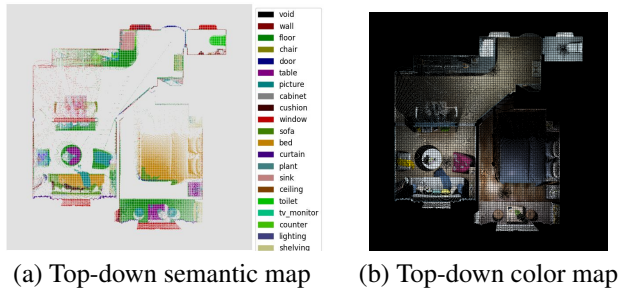


Figure 3. Visualization of a Structured3D scene: (a) Top-down semantic map with Matterport categories from the GeoSem Map, and (b) Top-down color map representing visual information.

scene using the A* algorithm. The agent maintains a ‘predicted map’ \mathbf{P} initialized with all cells marked as unobserved (0.5). At each timestep t , the agent simulates a depth sensor (LIDAR) to gather occupancy information within its specified FOV and orientation. The agent collects depth data up to the maximum ‘sensor range’ distance, updating \mathbf{P}_t accordingly. The observations at each timestep are derived from the ground truth occupancy map \mathbf{C} ,

$$\mathbf{P}_{t+1} = \mathbf{P}_t \odot (1 - \mathbf{O}_t) + \mathbf{C} \odot \mathbf{O}_t, \quad (11)$$

where \mathbf{O}_t is the observation mask at timestep t indicating the regions observed by the agent. Figure 4 illustrates an overview of the inputs, outputs, and ground truths for the Imagination Module.

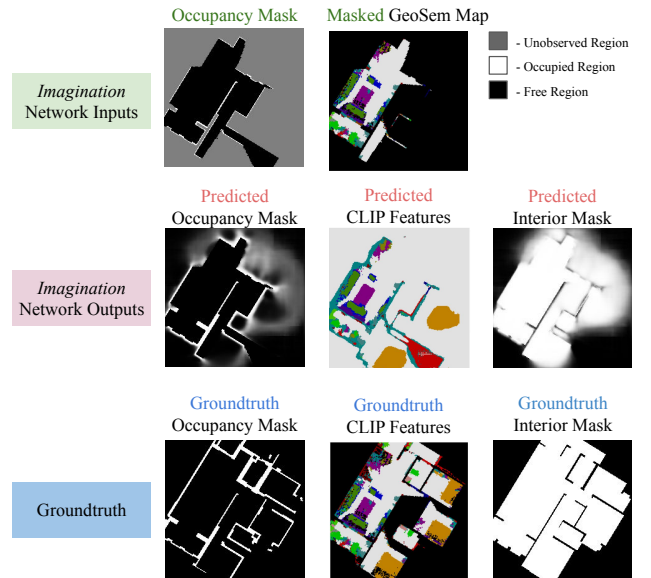


Figure 4. Example of inputs, predictions, and the associated ground truths for the Imagination Module. The GeoSem Map is a multi-dimensional tensor, but for visualization purposes, we show the top-down semantic map generated using the Matterport object categories as queries.

Network Architecture and Training Setup: To make the training process robust to various real-world scenarios, we apply on-the-fly data augmentations such as rotation and scaling to the training samples. We explore two different architectures for the imagination network: a Vision Transformer (ViT) [10] based architecture using Masked Autoencoders (MAE) and a Convolutional Neural Network (CNN) based architecture using a U-Net [31]. We supervise the imagination network using three loss components: CLIP embedding similarity, occupancy prediction, and interior mask prediction.

For *CLIP Embedding Similarity*, we use the cosine similarity loss for the CLIP embeddings, applied only to the in-

terior regions since it will be undefined for exterior regions with the CLIP-Embeddings set to zero vectors,

$$\mathcal{L}_{\text{CLIP}} = \frac{1}{|\Omega_{\text{int}}|} \sum_{(i,j) \in \Omega_{\text{int}}} \left(1 - \frac{\mathcal{G}_{i,j,1:D} \cdot \mathcal{J}_{i,j,1:D}}{\|\mathcal{G}_{i,j,1:D}\| \|\mathcal{J}_{i,j,1:D}\|} \right),$$

where Ω_{int} is the set of interior grid cells, \mathcal{J} is the f_{imagine} output and \mathcal{G} is the groundtruth. Since the exterior regions of the scene do not receive any supervision, the network tends to output random CLIP-embedding values for these regions. To ensure that these do not affect the long-term navigation goal selection, we also predict the interior mask and only look for the navigation goal within the predicted interior mask.

The occupancy prediction loss is defined as

$$\mathcal{L}_{\text{occ}} = \text{wBCE}(\mathcal{J}_{i,j,D+1}, \mathcal{G}_{i,j,D+1}), \quad (12)$$

which is a weighted binary cross-entropy (BCE) loss to account for the sparsity of occupied cells as compared to unoccupied cells. We also ablate by treating occupancy prediction as a regression problem by using the MSE loss for supervision. The interior mask is supervised using a standard BCE loss $\mathcal{L}_{\text{interior}} = \text{BCE}(\mathcal{J}_{i,j,D+2}, \mathcal{G}_{i,j,D+2})$. Thus, the final formulation of the multi-task loss is as

$$\mathcal{L} = \lambda_{\text{CLIP}} \cdot \mathcal{L}_{\text{CLIP}} + \lambda_{\text{occ}} \cdot \mathcal{L}_{\text{occ}} + \lambda_{\text{interior}} \cdot \mathcal{L}_{\text{interior}}. \quad (13)$$

3.3. Long-Term Navigation Goal Extraction

The task of exploration during ObjectNav can be summarized as choosing the point (x, y, z) to navigate to, given the past observations, so that the target object can be found. In our approach, we use the predicted GeoSem-Map output from f_{imagine} to decide the navigation goal. Since we make use of CLIP features, we can use any language query to navigate to, and the query does not need to be from a specified set of target categories. Given $\mathcal{J} = f_{\text{imagine}}(\mathcal{M}_t)$, where \mathcal{M}_t is the GeoSem Map constructed from observations from $(0, t]$, we choose the navigation goal using the cosine similarity heatmap between $\mathcal{J}[\mathcal{J}_{:, :, D+2} > 0.5]$ and $\text{CLIP}_{\text{text}}(\text{Goal_Query})$ - only for the pixels that lie inside the interior mask. Ideally, the pixel in the heatmap with the highest score should be the goal. To minimize false positives and ensure the robustness of the goal location, and to account for multiple occurrences of the query, we process the heatmap further. We first normalize and threshold the similarity map to get a binary heatmap indicating potential goal locations. Next, we apply some outlier removal using DBSCAN and find the optimal number of clusters (K) using the silhouette score metric. We then fit a Gaussian Mixture Model (GMM) with K gaussians to get the final candidate zones for the goal query, visualized in Figure 5. In Figure 6, the performance on an open-vocabulary language

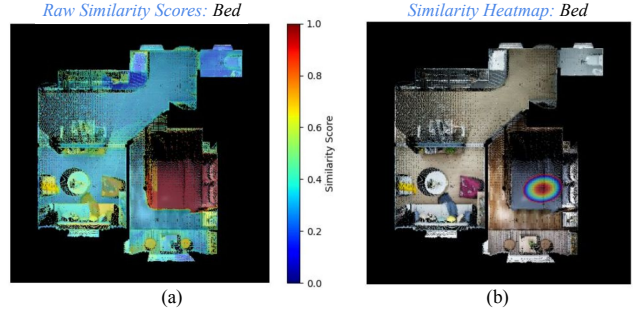


Figure 5. Visualization of query spatial distribution: (a) Raw similarity scores heatmap overlaid on a top-down RGB map showing potential locations of the query. (b) Gaussian Mixture Model (GMM) covariances as ellipses and log-likelihood heatmap overlaid on the top-down color map, visualizing the spatial distribution of the query.



Figure 6. An example of querying an open-vocabulary language query in a GeoSem Map: "Place for a Coffee Chat". On the left, we show the heatmap with GMM covariances and on the right the true locations of the query in the scene.

query is visualized. We convert the coordinates of the location from grid to world for the planner, and we choose the closest cluster to the current robot position when $K > 1$.

Imagination and goal selection are triggered at the same frequency - whenever a new *GeoSem Map* is predicted, a new navigation goal is chosen. The map is updated when the agent's current field of view has low overlap with the field of view at the last update, based on a predefined threshold.

4. Experiments and Results

Dataset: For the training of the Imagination module and evaluation of our method, we use scenes from the Structured3D dataset [41], which is a large-scale photo-realistic dataset with 3500 scenes. We train the Imagination module on the train-val split recommended by the dataset authors i.e. 3000 / 500. Each room in a scene has panoramic RGB and depth images, semantic annotations, and ground truth junction information, which makes it easy to construct a floorplan of the room from the annotations, providing

the ground truth occupancy map for walls. This makes Structured3D an ideal candidate for training the imagination layer of our exploration strategy. With this work we introduce the Structured3D benchmark for ObjectNav. All evaluations are reported on the validation split of Structured3D, which we do not use for training the Imagination Module.

Network Details: We use $H = 224, W = 224$ as the spatial dimensions for the GeoSem Map, with $z_{\min} = 30\%$ and $z_{\max} = 70\%$ of the scene height to emulate an occupancy map for a drone agent. The ViT-B/32 model variant of CLIP is used, which gives us $D = 512$ dimensional embeddings, making the size of our GeoSem Map $224 \times 224 \times 513$. The convolution depths for the UNet architecture are [64, 128, 256, 512, 1024] and the threshold to convert the predicted occupancy probability to a binary occupancy map for navigation is set to 0.72. We also ablate on using the *mae_vit_base_patch16* [18] ViT model for the Imagination module. The loss coefficients were set as $\lambda_{\text{CLIP}} = 10, \lambda_{\text{occ}} = 1, \lambda_{\text{interior}}=1$.

Evaluation Setup: The Structured3D is set up for the closed-loop evaluation task of ObjectNav as defined in Section 3. To evaluate the performance of ObjectNav, we employ three key metrics as proposed by [1]: Success, Success weighted by Path Length (SPL), and Distance to Goal. An agent is randomly initialized in a grid world, and its observations are simulated using the groundtruth GeoSem Map of the scene, which has been precomputed. The simulation for this evaluation occurs in a similar manner to the generation of occupancy masks in Section 3.2, where an agent navigates a 2D grid world using predefined parameters such as field of view (fov), sensor range, and turn angle.

4.1. Evaluating on Occupancy Prediction

Before we evaluate our approach on ObjectNav, we experiment with the Imagination strategy on the task of PointNav (Point Goal Navigation), where the task is to reach a point-goal (x, y, z) in an unseen environment. The agent starts in a random position and it is given two goal waypoints (x_1, y_1) and (x_2, y_2) to reach in order, with the simulation running for a maximum of 3000 timesteps. We compare the performance of two agents, both using the A* algorithm for planning:

1. *Imagination Agent:* Every 10 timesteps, the imagination agent imagines the occupancy map using the imagination layer and the A* planner uses this imagined map to plan its path to the goal.
2. *Vanilla Agent:* The vanilla agent replans using the current occupancy map every 10 timesteps.

In Table 1, we report each method’s completion rate and mean timesteps. We based our analysis on the network architecture and the loss we used to supervise the Imagina-

tion Module by treating occupancy prediction as a regression(MSE Loss) and classification(BCE Loss) task.

Method	Completion Rate (\uparrow)	Mean (\downarrow)
Vanilla	0.99	640.65
UNet + BCE	1.0	439.38
UNet + MSE	1.0	448.23
ViT + BCE	0.85	465.72
ViT + MSE	0.89	499.78

Table 1. Closed-loop simulation evaluation results. Arrows indicate whether higher (\uparrow) or lower (\downarrow) values are better. Bold values indicate the best performance for each metric.

The imagination agents outperform the vanilla agent with the UNet Imagination backbone supervised using BCE Loss, which gives the best performance, providing close to a **1.5x speedup** in reaching the goals compared to the vanilla agent. This result shows that predicting the walls from partially observed occupancy maps helps the agent navigate faster. Figure 7 visualizes the paths taken by the vanilla agent and the imagine agent for a particular run. After reaching the first waypoint, the imagine agent accurately predicts the structure of the scene and chooses the path from the south, which is closer to the ground truth best path, unlike the vanilla agent that attempts to reach the goal from the north.

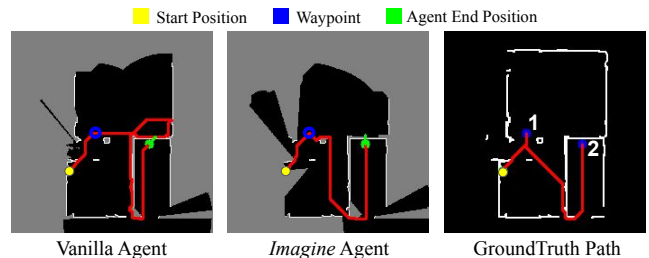


Figure 7. Paths taken by the vanilla agent and the imagine agent during a run. The imagine agent predicts the structure of the scene and selects a more efficient path. In the first two images, the green circle denotes the final position of the agents after reaching the goals. The last image (groundtruth path) shows the starting position of the agent in green, and the two goal waypoints in blue.

This observation backs our hypothesis that employing an approach backed by human reasoning for exploration can yield substantial gains in navigation efficiency. The ability of the imagine agent to anticipate the layout of unobserved regions and adapt its path accordingly demonstrates the potential of integrating imagination layers in navigation tasks.

4.2. Evaluating Imagination for ObjectNav

We evaluate our Imagination exploration strategy against other recent baselines. We choose frontier-based exploration techniques as baselines for a fair evaluation since our method is modular-learning based. Defined in [15], modular-learning approaches use an intermediate scene representation to compute the navigation goal as opposed to end-to-end approaches that directly make predictions in the action space. We evaluate and report metrics for each method for each category of the ObjectNav task as described in [1], namely: [“chair”, “couch”, “potted plant”, “bed”, “toilet”, “tv”] with an episode timeout of 500s. For all agents, when the goal coordinate appears in their observation, i.e., $P[g_x, g_y] \neq 0.5$ where P is the current occupancy map, it is assumed that the agent can detect the goal category and the exploration phase is over and the agent deterministically navigates to the goal.

Baselines: All baselines are frontier-based, and hence, at each timestep, the agent needs to decide which frontier to choose for exploration, which becomes the navigation goal for the planner.

1. *VLFM-CLIP:* VLFM [36] agent using CLIP for language grounding of features instead of BLIP-2 [23].
2. *StructNav-Frontiers:* Inspired by the approach of StructNav [7], this baseline uses a top-down semantic map with K fixed object categories (as introduced in [6]) as the scene representation instead of scene graphs proposed in the original method. The agent uses a prior distance matrix - a $K \times K$ matrix that stores similarity scores between the CLIP-Embedding vectors of the K classes. The frontier with the highest score indexed from the prior matrix using the semantic category at the frontier and the goal category, is chosen as the navigation goal.
3. *Random Agent:* Chooses a random frontier as the navigation goal in each timestep.
4. *Greedy Agent:* Chooses the frontier that is closest to its current position at the current timestep.

Method	SPL (\uparrow)	Distance To Goal (\downarrow)	Success (\uparrow)
ForesightNav (Ours)	0.67	25.32	0.73
VLFM-CLIP	0.66	27.36	0.71
StructNav-Frontiers	0.63	33.38	0.66
Random Agent	0.62	29.30	0.68
Greedy Agent	0.58	37.06	0.60

Table 2. Average metrics for different exploration strategies in the ObjectNav task. Our proposed approach outperforms existing state-of-the-art approaches on the Structure3D validation dataset in three different evaluation metrics.

We report ObjectNav metrics averaged across the vali-

ation scenes of the Structured3D dataset for each of the ObjectNav categories in Table 2.

By leveraging imagination-driven exploration, **ForesightNav** achieves absolute increases in the success rates weighted by path length (SPL) over prior SOTA approaches. This improvement is driven by its ability to anticipate unseen regions, effectively guiding the agent towards high-reward areas without unnecessary exploration. This suggests that imagination aided by GeoSem Maps enhance the agent’s ability to infer promising frontiers, prioritizing those most likely to contain the target object. This capability is particularly valuable in large or complex environments where exhaustive search is computationally expensive.

While StructNav-Frontiers also leverages semantic information, it is outperformed by ForesightNav and VLFM-CLIP due to the semantic richness that CLIP embeddings provide. The Random Agent and Greedy Agent predictably perform the worst, further emphasizing the critical role of advanced scene representation and the exploration strategy in efficient navigation.

5. Conclusion

In this work, we have presented *ForesightNav*, a novel exploration strategy for Object Goal Navigation that leverages the power of data-driven imagination. By integrating CLIP’s visual-semantic understanding with an anticipatory reasoning framework, our method enables an agent to infer unexplored regions and make informed navigation decisions, leading to superior performance over prior SOTA approaches. Our results demonstrate that *ForesightNav* significantly improves navigation efficiency for both the PointNav and ObjectNav tasks, also achieving higher success rates.

Our findings suggest that data-driven imagination is the next step in robotic exploration, with the potential to redefine how agents navigate complex environments. By training agents to *imagine* spaces before seeing them, we move closer to creating foundation models for indoor scene understanding, similar to how large vision-language models have revolutionized perception.

Limitations and Future Work: While the imagination strategy we propose is effective in indoor environments with a BEV-style GeoSem map, for navigating large scenes using non-object-centric language queries - a higher resolution map must be used, which will significantly increase memory usage and the network size of the imagination model. A hierarchical memory structure that builds on top of GeoSem Maps and merges it with a structure like 3D Scene Graphs [2] for faster retrieval is a possible extension to this work.

6. Acknowledgements

This work was supported by the European Union’s Horizon Europe Research and Innovation Programme under grant agreement No. 101120732 (AUTOASSESS) and the European Research Council (ERC) under grant agreement No. 864042 (AGILEFLIGHT).

References

- [1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 7, 8
- [2] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5664–5673, 2019. 8
- [3] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020. 2, 3
- [4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3
- [5] Matthew Chang, Arjun Gupta, and Saurabh Gupta. Semantic visual navigation by watching youtube videos. *Advances in Neural Information Processing Systems*, 33:4283–4294, 2020. 2
- [6] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020. 2, 8
- [7] Junting Chen, Guohao Li, Suryansh Kumar, Bernard Ghanem, and Fisher Yu. How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers. *arXiv preprint arXiv:2305.16925*, 2023. 2, 8
- [8] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Proctor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022. 2
- [9] Vishnu Sashank Dorbala, James F Mullen Jr, and Dinesh Manocha. Can an embodied agent find your “cat-shaped mug”? IIm-based zero-shot object navigation. *IEEE Robotics and Automation Letters*, 2023. 2
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5
- [11] Russell A Epstein, Eva Zita Patai, Joshua B Julian, and Hugo J Spiers. The cognitive map in humans: spatial navigation and beyond. *Nature neuroscience*, 20(11):1504–1513, 2017. 1
- [12] Ludvig Ericson and Patric Jensfelt. Beyond the frontier: Predicting unseen walls from occupancy grids by learning from floor plans. *IEEE Robotics and Automation Letters*, 2024. 2, 3
- [13] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23171–23181, 2023. 2
- [14] Sourav Garg, Krishan Rana, Mehdi Hosseinzadeh, Lachlan Mares, Niko Sünderhauf, Feras Dayoub, and Ian Reid. Robohop: Segment-based topological map representation for open-world visual navigation. *arXiv preprint arXiv:2405.05792*, 2024. 2
- [15] Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. Navigating to objects in the real world. *Science Robotics*, 8(79):eadf6991, 2023. 3, 8
- [16] Nandiraju Gireesh, DA Sasi Kiran, Snehasis Banerjee, Mohan Sridharan, Brojeshwar Bhowmick, and Madhava Krishna. Object goal navigation using data regularized q-learning. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 1092–1097. IEEE, 2022. 2
- [17] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv preprint arXiv:2309.16650*, 2023. 2
- [18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 7
- [19] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023. 3, 4
- [20] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023. 2
- [21] Kapil Katyal, Katie Popek, Chris Paxton, Phil Burlina, and Gregory D Hager. Uncertainty-aware occupancy map prediction using generative networks for robot navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5453–5459. IEEE, 2019. 2, 3
- [22] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 2, 3

- [23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 8
- [24] Yimeng Li, Arnab Debnath, Gregory J Stein, and Jana Košecká. Learning-augmented model-based planning for visual exploration. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5165–5171. IEEE, 2023. 2
- [25] Balmukund Mishra, Deepak Garg, Pratik Narang, and Vipul Mishra. Drone-surveillance for search and rescue in natural disaster. *Computer Communications*, 156:1–10, 2020. 1
- [26] Kaixian Qu, Jie Tan, Tingnan Zhang, Fei Xia, Cesar Cadena, and Marco Hutter. Ippon: Common sense guided informative path planning for object goal navigation. *3rd Workshop on Language and Robot Learning: Language as an Interface*, 2024. 2
- [27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3
- [28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2
- [29] Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17896–17906, 2023. 2
- [30] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. In *7th Annual Conference on Robot Learning*, 2023. 2
- [31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 5
- [32] Antonin Vobecky, Oriane Siméoni, David Hurych, Spyridon Gidaris, Andrei Bursuc, Patrick Pérez, and Josef Sivic. Pop-3d: Open-vocabulary 3d occupancy prediction from images. In *Advances in Neural Information Processing Systems*, pages 50545–50557. Curran Associates, Inc., 2023. 3
- [33] Jiaxu Xing, Giovanni Cioffi, Javier Hidalgo-Carrió, and Davide Scaramuzza. Autonomous power line inspection with drones via perception-aware mpc. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1086–1093. IEEE, 2023. 1
- [34] Brian Yamauchi. 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*, pages 146–151. IEEE, 1997. 2
- [35] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectnav. *arXiv preprint arXiv:2104.04112*, 2021. 2
- [36] Naoki Harrison Yokoyama, Sehoon Ha, Dhruv Batra, Jiguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023. 2, 8
- [37] Kuo-Hao Zeng, Zichen Zhang, Kiana Ehsani, Rose Hendrix, Jordi Salvador, Alvaro Herrasti, Ross Girshick, Aniruddha Kembhavi, and Luca Weihs. Poliformer: Scaling on-policy rl with transformers results in masterful navigators. *Conference on Robot Learning*, 2024. 2
- [38] Mike Zhang, Kaixian Qu, Vaishakh Patil, Cesar Cadena, and Marco Hutter. Tag map: A text-based map for spatial reasoning and navigation with large language models. *Conference on Robot Learning*, 2024. 2
- [39] Sixian Zhang, Xinhang Song, Yubing Bai, Weijie Li, Yakui Chu, and Shuqiang Jiang. Hierarchical object-to-zone graph for object navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15130–15140, 2021. 2
- [40] Sixian Zhang, Xinyao Yu, Xinhang Song, Xiaohan Wang, and Shuqiang Jiang. Imagine before go: Self-supervised generative map for object goal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16414–16425, 2024. 2, 3
- [41] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 519–535. Springer, 2020. 6, 2
- [42] Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *International Conference on Machine Learning*, pages 42829–42842. PMLR, 2023. 2

ForesightNav: Learning Scene Imagination for Efficient Exploration

Supplementary Material

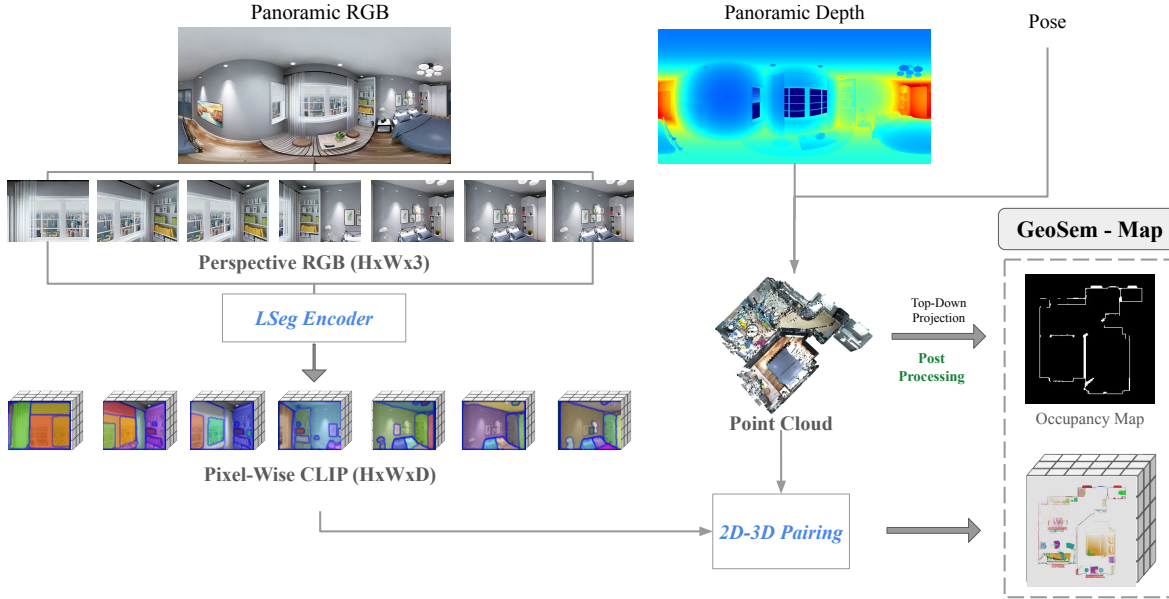


Figure 8. GeoSem Map generation for the Structured3D Dataset (Equirectangular RGBD)

7. GeoSem Maps from Equirectangular RGBD + Pose

The Structured3D dataset [41] provides richly annotated 3D indoor scenes with panoramic RGBD observations. It contains 3,500 scenes in total, split into 3,000 for training and 500 for validation. Each scene S_i is composed of multiple rooms, and for every room R_{ij} , the dataset offers a set of equirectangular RGB images $I_{\text{equirect}_{ij}} \in \mathbb{R}^{H \times W \times 3}$, corresponding equirectangular dense depth maps $D_{\text{equirect}_{ij}} \in \mathbb{R}^{H \times W \times 1}$, and global camera poses $T_{ij} \in SE(3)$ for each panoramic viewpoint.

In this section, we describe the construction of *GeoSem Maps* for each scene, which serve as groundtruth supervision for the imagination module discussed in Section 3.2. A representative Figure 8 depicts the overview of the method. Directly using the equirectangular images I_{equirect} as input to the pre-trained LSeg encoder leads to degraded CLIP embeddings due to the geometric distortions introduced by the panoramic format. To mitigate this, we instead apply a perspective projection strategy, simulating standard pin-hole camera views to generate high-quality, per-pixel CLIP embeddings.

7.1. Perspective Projection for GeoSem Map Construction

We utilize a perspective camera model with a field of view (FOV) of 90 degrees. The viewing directions are strategically chosen so that the complete room is covered: horizontal view directions $\Theta = [-180^\circ, -90^\circ, 0^\circ, 90^\circ]$ and vertical view directions $\Phi = [-45^\circ, 0^\circ, 45^\circ]$. Each combination of Θ and Φ results in a distinct perspective image capturing the room from different orientations i.e. 12 viewing directions. Given an equirectangular image $I_{\text{equirect}_{ij}}$ of scene S_i and room R_{ij} , perspective projection gives us 12 images $I_{\text{persp}_{ijk}}$ where $k \in [1, 12]$. With perspective projection, we lose the pixel-wise depth association between I_{equirect} and D_{equirect} . To finally construct a GeoSem Map as described in 3.1, we need pixel-wise CLIP embeddings and depth, along with camera pose for top-down projection onto a BEV map.

7.2. 2D-3D Correspondence Using Indexing Approach

Each pixel (u, v) in $D_{\text{equirect}_{ij}}$ represents a depth value d_{uv} , which can be projected into a local 3D point $\mathbf{p}_{uv}^{\text{local}} \in \mathbb{R}^3$ using the intrinsic parameters of the equirectangular projection. These local 3D points are then transformed into the global coordinate frame using the camera pose:

$$\mathbf{p}_{uv}^{\text{global}} = T_{ij} \cdot \mathbf{p}_{uv}^{\text{local}}$$

A global point cloud is constructed for the scene by aggregating all such transformed 3D points across rooms:

$$\mathcal{P}_i = \bigcup_j \{\mathbf{P}_{uv}^{\text{global}} \mid (u, v) \in D_{\text{equirect}_{ij}}\}$$

To retain the pixel-to-point correspondence for associating the CLIP embeddings later, an indexing scheme is used:

- **Base Indexing:** Each panoramic image $I_{\text{equirect}_{ij}}$ is assigned a unique base index B_{ij} .
- **Pixel Indexing:** We create the index map $\mathbf{index}_{ij} \in \mathbb{R}^{H \times W \times 1}$, where each pixel (u, v) receives a unique global index within the scene:

$$\mathbf{index}_{ij}(u, v) = B_{ij} + (u \cdot W + v)$$

This ensures that every pixel across all panoramas in a scene has a globally unique identifier.

Perspective Projection and Index Transformation:

When we convert $I_{\text{equirect}_{ij}}$ into a perspective image $I_{\text{persp}_{ijk}}$, we similarly project the index map \mathbf{index}_{ij} to obtain a corresponding *index perspective image* ϕ_{ijk} which allows us to define the mapping:

$$\begin{aligned} \phi_{ijk} &: \text{Pixels in } I_{\text{persp}_{ijk}} \rightarrow \text{Point IDs in } \mathcal{P}_i \\ \phi_{ijk}(u', v') &= \mathbf{index}_{ij}(u, v) \end{aligned}$$

This enables each pixel in the perspective image to retain the identity of its original 3D point, thereby preserving the connection between CLIP embeddings (obtained from LSeg) and their corresponding 3D positions in the scene.

$$\text{LSeg}(I_{\text{persp}_{ijk}})(u', v') \rightarrow \mathcal{P}_i[\phi_{ijk}(u', v')]$$

7.3. CLIP Embedding Generation and Mapping

With the perspective images and corresponding depth information associated using the index images, we employ the LSeg encoder to derive pixel-wise CLIP embeddings for each perspective image.

The subsequent steps involve aggregating these CLIP embeddings into a grid representation \mathcal{M} , where each grid cell encapsulates semantic information from the underlying 3D points and their associated CLIP embeddings. The construction of the GeoSem Map \mathcal{M} proceeds as outlined in Section 3.1.

7.4. Handling Spurious Observations

From the Structured3D annotations, the groundtruth floor plan is known. However, to better simulate real-world conditions during training, we generate the groundtruth occupancy maps using a top-down projection of the global point-cloud instead of using the dataset floor plan annotation. Due to imperfections in the point cloud, the resulting occupancy

maps may contain small holes in the walls. These can cause "leaks" during agent simulation (described in Section 4) where depth rays may erroneously pass through walls. This leads to artifacts in the occupancy map - specifically, observed regions outside the intended scene boundaries which would not occur in a realistic setting. To address this, we generate an interior-exterior mask $\mathbf{E} \in \{0, 1\}^{M \times M}$ from the Structured3D annotations for room polygons, marking all points in the exterior as 0.5 (unobserved) in the occupancy map P :

$$\mathbf{P} = \mathbf{P} \odot \mathbf{E} + 0.5(1 - \mathbf{E}) \quad (14)$$