

VISTA: Open-Vocabulary, Task-Relevant Robot Exploration with Online Semantic Gaussian Splatting

Keiko Nagami¹, Timothy Chen¹, Javier Yu¹, Ola Shorinwa¹, Maximilian Adang¹,
Carlyn Dougherty², Eric Cristofalo², and Mac Schwager¹

Abstract—We present VISTA (Viewpoint-based Image selection with Semantic Task Awareness), an active exploration method for robots to plan informative trajectories that improve 3D map quality in areas most relevant for task completion. Given an open-vocabulary search instruction (e.g., “find a person”), VISTA enables a robot to explore its environment to search for the object of interest, while simultaneously building a real-time semantic 3D Gaussian Splatting reconstruction of the scene. The robot navigates its environment by planning receding-horizon trajectories that prioritize semantic similarity to the query and exploration of unseen regions of the environment. To evaluate trajectories, VISTA introduces a novel, efficient viewpoint-semantic coverage metric that quantifies both the geometric view diversity and task relevance in the 3D scene. On static datasets, our coverage metric outperforms state-of-the-art baselines, FisherRF and Bayes’ Rays, in computation speed and reconstruction quality. In quadrotor hardware experiments, VISTA achieves 6x higher success rates in challenging maps, compared to baseline methods, while matching baseline performance in less challenging maps. Lastly, we show that VISTA is platform-agnostic by deploying it on a quadrotor drone and a Spot quadruped robot. Code and videos can be found on our project page: <https://stanfordmsl.github.io/VISTA/>.

I. INTRODUCTION

Research advances in vision and language foundation models, e.g., [1], [2], have enabled language-guided object localization in pre-mapped real-world environments [3]. For example, a user can task a robot with the word *apple* and it will find an apple in a pre-mapped concept graph [4]. However, to find query objects efficiently in *unstructured, unmapped environments*, robots must be capable of exploring their environments intelligently, with a bias toward finding the object of interest. Prior work in robot exploration broadly uses

Manuscript received: June, 17, 2025; Revised November, 16, 2025; Accepted November, 29, 2025.

This paper was recommended for publication by Editor Olivier Stasse upon evaluation of the Associate Editor and Reviewers’ comments.

¹Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA knagami, chengine, javieryu, shorinwa, madang, schwager@stanford.edu.

²MIT Lincoln Laboratory, Lexington, MA 02421, USA carlyn.dougherty, eric.cristofalo@ll.mit.edu.

This work was supported in part by MIT Lincoln Laboratory ACC grant 7000603941, NSF project FRR 2342246, and ONR project N00014-23-1-2354. The NASA University Leadership initiative (grant #80NSSC20M0163) provided funds to assist the authors with their research, but this article solely reflects the opinions and conclusions of its authors and not any NASA entity.

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the Department of the Air Force under Air Force Contract No. FA8702-15-D-0001 or FA8702-25-D-0002. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Air Force. ©2025 Massachusetts Institute of Technology. Delivered to the U.S. Government with Unlimited Rights, as defined in DFARS Part 252.227-7013 or 7014 (Feb 2014). Notwithstanding any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work.

Digital Object Identifier (DOI): see top of this page.

©2026 IEEE

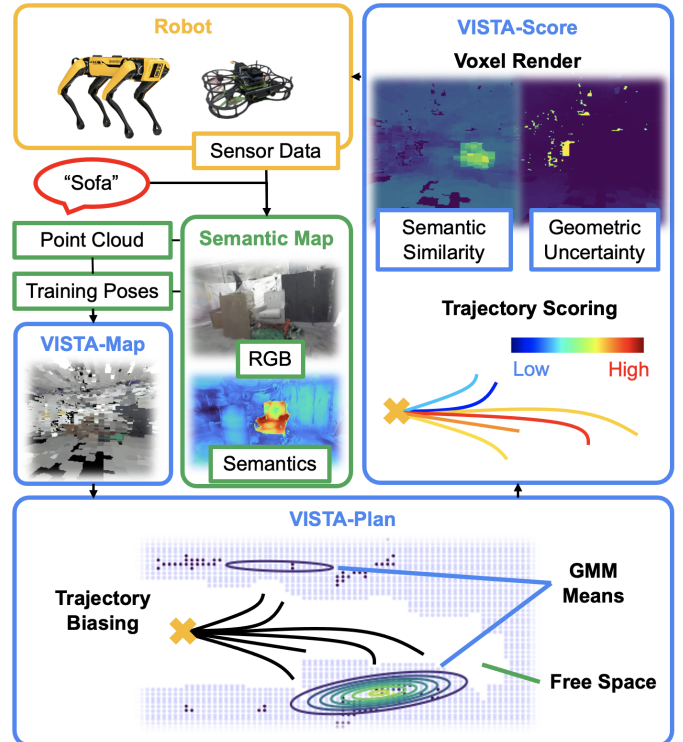


Fig. 1. System overview of VISTA. Real-time onboard sensor data is gathered to train a semantic 3DGS map. The semantic and RGB information from the 3DGS map are transferred to a 3D Voxel Grid, and training poses are used to store geometric information about directions from which each voxel has already been viewed. Frontier cells and semantic information are used to fit a GMM that is sampled to generate candidate trajectories. The trajectory with the highest semantic + geometric information gain is then executed in a receding horizon loop.

traditional 3D scene representations, such as occupancy and voxel grids. We build upon these traditional representations by introducing a Gaussian Splat embedded with semantic information that can advance downstream tasks to be performed by the robot.

We present VISTA, an algorithm for Viewpoint-based Image Selection with Semantic Task Awareness. To enable task-relevant exploration, VISTA introduces two key innovations: (i) a semantics-aware mapping and information-gain pipeline that leverages open-vocabulary semantics for task-relevant exploration, and (ii) a scalable information-gain metric based on view angle diversity that can be computed efficiently in real-time. Our key insight is that, for vision-based mapping, the variety and multiplicity of viewing directions from which an environment point has appeared in the image history is a strong proxy for the geometric quality of the reconstruction at that point.

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

First, VISTA builds a high-fidelity photorealistic map of the robot’s environment online using a Gaussian Splatting (3DGS) representation [5]. To enable open-vocabulary, task-relevant robot exploration, VISTA distills semantic features from vision-language models, e.g., CLIP [1], into the 3DGS map incrementally as new observations are obtained by the robot. The semantic features encode the relevancy between each point in the environment and the specified exploration task, giving a task relevancy 3D heatmap. By using 3DGS, our map representation has reduced noise and additional semantic cues from RGB images that depth information alone would be unable to provide. In comparison to other high fidelity scene representations such as neural radiance fields (NeRF) [6], 3DGS also provides faster training times that are essential in a real-time exploration task where the robot must move while asynchronously updating its map.

Using the 3DGS map, VISTA simultaneously constructs a voxel grid, capturing the viewed regions of the scene and the semantic relevance of these regions. With this grid, we measure view diversity through a conceptually simple, computationally efficient coverage metric with advantages like recursive updating and informative trajectory optimization. In each voxel, the geometric uncertainty is the minimum angular separation between the test viewpoint and all view angles from which that voxel has appeared in the image history accounting for occlusions.

Finally, VISTA samples trajectories, and selects those with viewpoints that maximize a weighted combination of geometric uncertainty and semantic relevance, ultimately guiding the robot toward a specified query object. We illustrate these components in Figure 1. Through an experimental campaign with a total of 36 hardware executions, we show that VISTA outperforms state-of-the-art baselines, achieving 6x better success rates in environments where the object is not visible from the initial robot pose, and matching performance when it is visible.

Our contributions are as follows. We introduce:

- 1) an efficient information metric that combines view angle diversity and semantic task relevance stored on a voxel grid that can be recursively updated,
- 2) a real-time informative trajectory planning algorithm to drive robot exploration using this metric,
- 3) and a full stack ROS implementation of VISTA and online 3DGS training demonstrated on robot hardware.

II. RELATED WORK

Robot Exploration. The objective in robot exploration is to traverse through an environment efficiently to build a complete map. The maps used in these problems have traditionally used occupancy grids, point clouds, signed distance fields (SDFs), and voxel maps. Within these map environments, exploration is typically performed through frontier-based, or sampling-based methods. Frontiers are defined within the map as boundaries between free and unknown space, where the robot can collect observations of the unknown regions [7]–[10]. In viewpoint sampling methods, each view is scored by an information metric where the highest scoring are prioritized

[11]–[14]. These two methods are often used together, in order to improve sample efficiency of the candidate viewpoints [15], [16].

Building upon these methods, VISTA incorporates both frontier-based and viewpoint sampling methods for its exploration pipeline, using a high-fidelity 3DGS map representation, incorporating semantic information in both its map and trajectory scoring to search for objects in the scene.

Active Planning and View Selection in Radiance Fields. Recently, radiance field methods, such as NeRFs [6] and 3DGS [5] have been introduced to robot mapping [17]–[19], to generate high fidelity representations of environments. Of these methods, 3DGS provides much faster training and rendering rates, enabled by an explicit, interpretable representation of the scene, and a tile-based rasterization procedure, which is more efficient than the volumetric ray-marching procedure used by NeRFs.

Active planning algorithms use view selection methods in order to gather the most information to improve localization accuracy [20]–[27] and to estimate the uncertainty of radiance fields via the Fisher information metric [25], [26], [28] or perturbation-based analysis [29]. However, these methods are generally not used directly for active mapping. RT-GuIDE [30], which is most relevant to our work, computes the magnitude of the change in the parameters of a 3DGS map over consecutive updates to estimate the uncertainty of the map in active exploration, with hardware demonstrations on a robot. Unlike VISTA, these methods make stringent assumptions on the map representations and lack semantic information, hindering their generality.

Semantics-Guided Exploration. Semantic radiance fields [31]–[36] enable open-vocabulary object search and exploration [37]–[42], by distilling semantic information from 2D vision language foundation models, e.g., CLIP [1]. Broadly, many existing approaches [37] employ a switching-based control architecture that switches between exploration and exploitation based on geometry-based metrics [38]; however, defining a suitable switching rule is often challenging, especially in practical settings. Other methods [39] decompose the exploration task using local 3DGS submaps, without accounting for geometric uncertainty, which could struggle in scenarios without the target object. In contrast to these methods, VISTA integrates both semantics and geometric guidance into the exploration task robustly without relying on switching behavior modes, addressing these limitations.

III. PROBLEM FORMULATION

We consider an exploration problem in which a robot has an onboard, forward-facing RGB-D camera with reliable state estimation. The robot is placed into a previously unseen environment and is given an open-vocabulary query to locate and retrieve a certain object in the scene by a user. Once the robot receives the input query, it must then construct a map of its environment as it moves, while simultaneously searching for the query object. In this informative planning task, the robot must balance finding the object while generating a map during exploration to confidently determine whether or not the identified object satisfies the user’s input query.

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

To train the 3DGS and render images in the voxel grid, the camera pose of the robot’s onboard camera is represented as: $\mathbf{x} = [x \ y \ z \ \phi \ \theta \ \psi]^T$, including position and Euler angles of the camera in the global frame. As the robot moves, it collects full pose odometry along with RGB and depth images to train a 3DGS map of the environment. 3DGS [5] represents non-empty space using Gaussian primitives, each parametrized by a mean $\boldsymbol{\mu} \in \mathbb{R}^3$, a covariance $\boldsymbol{\Sigma} = \mathbf{RSS}^T\mathbf{R}^T \in \mathcal{S}_{++}^n$ (parameterized by a rotation matrix \mathbf{R} and a diagonal scaling matrix \mathbf{S}), an opacity $\alpha \in \mathbb{R}_+$, and spherical harmonics parameters to capture view-dependent visual effects like reflections. This explicit representation not only enables 3DGS to avoid unnecessary computation involving empty space, but also enables the utilization of fast tile-based rasterization. The rasterization procedure uses α -blending, computing the color of each pixel.

As the map updates, we assume that the motion of the robot is restricted in the ϕ , and θ axes. The robot’s motion is then modeled as a single integrator with a heading angle in the yaw direction. The state and control vectors for planning $\mathbf{s} \in \mathbb{R}^4$ and $\mathbf{u} \in \mathbb{R}^4$ are $\mathbf{s} = [x \ y \ z \ \psi]^T$, and $\mathbf{u} = [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\psi}]^T$, respectively, where the velocity control vector is subject to control limits.

IV. VISTA

We propose a method for efficient exploration of an environment by a robot, guided by a natural language semantic query provided by a user. Our robot constructs a 3DGS map in real-time as it moves, collecting odometry and RGB-D image information. We extract a voxel grid from the underlying radiance field for computational efficiency when computing information gain metrics. Within this voxelized representation, we distinguish between observed and unobserved space, where the observed space is further segmented into occupied and free space. To enable task-aligned scene coverage, we fuse geometric and semantic information-gain metrics, guiding the robot toward regions with high geometric uncertainty and semantic relevancy to the input query. Our method is visualized in Fig. 1.

A. Real-Time Semantic Radiance Field Training

To construct semantic 3DGS maps in real-time, VISTA builds upon NerfBridge [43], [44] and its 3DGS extension Splatbridge [45], both real-time methods for online training of radiance fields. In both, images from the robot’s onboard cameras and poses are aggregated into a streaming dataset that is used to continuously optimize the radiance field. To add semantic information to these platforms, we then distill semantic embeddings from the 2D vision-language model CLIP [1] into the online radiance field using the distillation procedure used in semantic NeRF literature, computing the CLIP image embeddings for each incoming image. Specifically, the trained semantic field $f: \mathbb{R}^3 \mapsto \mathbb{R}^l$, parametrized by a multi-resolution hashgrid followed by a multilayer perceptron, maps a 3D point to a semantic embedding.

To optimize the semantic field, 3D points back-projected from the predicted depth image are used to generate inputs for

training f . The parameters of f are optimized using gradient-based optimization of the mean-squared error (MSE) between the predicted and ground-truth CLIP semantic embeddings. We also include the cosine similarity as a component in the loss function in 3DGS. The parameters of f and those of the base 3DGS are trained simultaneously. We direct interested readers to [33], [34] for a more detailed discussion of training setup, performance, and trade-offs associated with semantic distillation methods for 3DGS. Prior work addresses comparisons between different semantic map representations [33], [46], [47]. We additionally compare our 3DGS approach against directly projecting CLIP information to a voxel grid in Fig. 4, and show that our method produces clearer distinctions between objects in the scene.

While the 3DGS map is training, a point cloud representation of the field containing RGB colors and semantic embeddings is published in real-time along with a subset of training poses. If the dataset size is below some maximum number of poses N , the full dataset is returned, otherwise N poses are sampled from the training dataset.

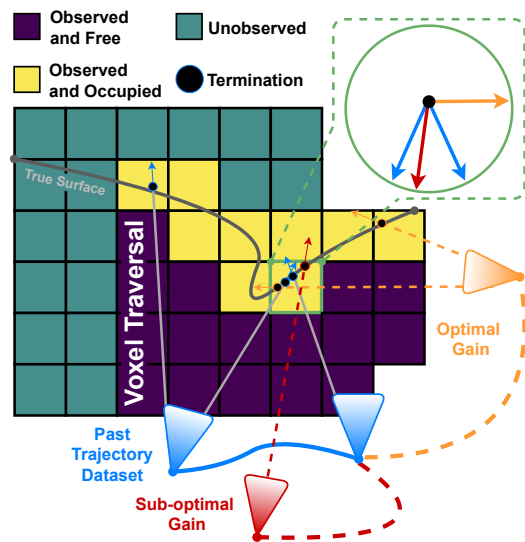


Fig. 2. Geometric information gain based on view diversity coverage. Voxel categories of free, occupied, and unobserved are determined through voxel traversal over the camera rays. All occupied voxels contain a list of view directions derived from the dataset (blue). Pixel-level geometric gain can be rendered through voxel traversal. A simple coverage metric is computed between the candidate view direction and the per-voxel list of dataset-derived directions. Viewpoints that are similar to the dataset render to a sub-optimal gain (red), while those that are different return a high gain (orange). The mechanism is visualized in the inset, with low gain directions being red and pointing to denser areas of the unit sphere, while high gain directions are orange and point towards sparser regions.

B. VISTA-Map: 3D Voxel Grid Representation

The published point cloud from the 3DGS training procedure is voxelized into a grid of fixed size and resolution centered on the robot’s current position. In this way, the map representation is restricted from growing with the number of Gaussians in the 3DGS map.

The voxels in the grid are characterized into one of three categories: occupied, free, and unobserved. All point cloud information (e.g. color and semantic embeddings) is registered into occupied voxels. To separate the remaining voxels into either unobserved or free cells, VISTA employs voxel traversal

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

[48] from the training cameras of the 3DGS pipeline. All rays corresponding to pixels of the training camera originate at the camera origin and terminate at an occupied voxel or exit the voxel grid. All voxels that intersect with rays up until termination are deemed free space. Remaining voxels are assigned as unobserved voxels.

Voxel traversal of the voxel map can similarly be used to render RGB, depth, semantic, and geometric gain images. From camera pose \mathbf{x} , rays parametrized as $\mathbf{r}(t) = \mathbf{o} + t \cdot \mathbf{d}$ corresponding to each pixel are cast into the voxel map, terminating when either an occupied or unobserved voxel is traversed or some maximum draw distance is reached. Rays that terminate at an occupied or unobserved voxel render that voxel's attributes into the pixel. We use the rendered images for active view planning in Section IV-C.

C. VISTA-Score: Information Gain Quantification

Next, we compute information gain metrics from the voxel grid. As described in Section IV-B, the training poses and the voxel traversal mechanism are used to store information about the directions from which camera rays of the training views intersect with the respective voxel. With this stored information, we evaluate the proposed geometric information gain of a candidate camera pose for a particular voxel by comparing the new proposed view direction with the existing view directions for that voxel. Specifically, we compute the dot product of each of the existing voxel's direction vectors against the new camera pose rays that intersect with that voxel. The dot product will approach -1 for new rays that contrast greatly from the closest training ray and 1 for those similar to the closest training ray. We then normalize this information between 0 and 1 for each ray from the proposed camera for the pixel-wise gain metric. Subsequently, we compute an image-wise geometric information gain for the candidate pose by taking the mean of the pixel values,

$$G_{\mathcal{I}}(\mathbf{x}) = \frac{1}{N_r} \sum_{n=1}^{N_r} \frac{\min(-\mathbf{d}_v^T \mathbf{d}_x^n) + 1}{2}, \quad (1)$$

where \mathbf{d}_x is an array of all direction vectors of rays generated from camera pose \mathbf{x} , superscript n denotes a single ray index in this array, and \mathbf{d}_v is an array of all direction vectors stored in voxel v from previous camera views. The min is taken over indices of the resulting vector, to give the cosine similarity between the proposed view and closest existing view, and N_r is the number of rays from the image taken at pose \mathbf{x} . For the semantic information gain, we also use the voxel traversal mechanism to produce a per-pixel semantic value in images rendered in the voxelized map. The semantic information gain is then computed as the mean value of all non-zero pixels in the image, averaged with the largest pixel value in the image. This term is denoted by $G_S(\mathbf{x})$, and the two metrics can be used together to compute a VISTA-Score along a trajectory of viewing directions as follows,

$$G(\bar{\mathbf{x}}) = \sum_{\mathbf{x} \in \bar{\mathbf{x}}} \gamma^{K-k} (cG_{\mathcal{I}}(\mathbf{x}) + G_S(\mathbf{x})), \quad (2)$$

where the path $\bar{\mathbf{x}}$ is described by a sequence of waypoints \mathbf{x} , K is the number of waypoints in the path, k is the index of the waypoint in the path, γ is a discount factor, and c is a weighting factor of the geometric information gain. In Section IV-D, we detail how paths are sampled for scoring.

D. VISTA-Plan: Informative Planning

To generate candidate paths for exploration (2), we sample trajectories that are biased toward regions of high information gain. The algorithm used for planning is detailed in Algo. 1. In this planning pipeline, we first create a 2D voxel grid, \mathcal{V}' from the 3D voxel grid, \mathcal{V} , by slicing a band of \mathcal{V} in the z -direction in which the robot will operate, omitting the floor and ceiling. The voxels are then assigned in \mathcal{V}' by priority in order of observed-occupied, unobserved, then observed-free, along the z -dimension. Semantic information is similarly encoded in \mathcal{V}' , by slicing a band of the 3D voxel grid, and summing the semantic values of each voxel grid cell in the height dimension. In Algo. 1, this procedure is captured in the function FlattenVoxelGrid.

Algorithm 1 VISTA-Plan

Require: $\mathbf{s}_i, \mathcal{V}, c, \beta, z$ **Output:** Best Trajectory $\bar{\mathbf{s}}^*$

- 1: $\mathcal{V}' = \text{FlattenVoxelGrid}(\mathcal{V})$
 - 2: $\mathcal{D}_f = \text{GetFrontiers}(\mathcal{V}')$
 - 3: $\mathcal{D}_s = \text{GetSemanticSamples}(\mathcal{V}')$
 - 4: $\mathcal{P} = \text{Dijkstra}(\mathbf{s}_i, \mathcal{V}')$
 - 5: $\hat{\mathcal{P}} \sim \text{SampleTrajectories}(\mathcal{P}, \text{GMM}(\mathcal{D}_f, \mathcal{D}_s))$
 - 6: $\mathcal{G} = \emptyset$
 - 7: **for** $\bar{\mathbf{p}} \in \hat{\mathcal{P}}$ **do**
 - 8: $\bar{\psi} = \text{FeasibleHeadings}(\mathcal{D}_f, \mathcal{D}_s, \bar{\mathbf{p}}, \mathbf{s}_i)$
 - 9: $\bar{\mathbf{s}} \leftarrow \bar{\mathbf{p}} \cup \bar{\psi}$
 - 10: $\bar{\mathbf{x}} \leftarrow \text{ConstructFullPose}(\bar{\mathbf{s}}, z)$
 - 11: $G = \text{VISTAScore}(\bar{\mathbf{x}}, \mathcal{V}, c)$
 - 12: $\mathcal{G} \leftarrow \mathcal{G} \cup G$
 - 13: **end for**
 - 14: $c = \beta^i c$, $\bar{\mathbf{s}}^* = \text{GetBestTrajectory}(\mathcal{G}, \hat{\mathcal{P}})$
-

Using this 2D representation of the environment, we encode global geometric and semantic information from the scene to bias trajectories toward regions with highest information gain. For geometric information gain, we search for frontier cells, \mathcal{D}_f from \mathcal{V}' , where observed-free cells are bordered by unobserved cells. For semantic information gain, the top- m 2D grid cells with the highest semantic similarity values are used to create a Categorical distribution that is then sampled to generate data, \mathcal{D}_s . The frontier cells along with the samples from the Categorical distribution are then used to fit a Gaussian Mixture Model (GMM) probability distribution across the environment.

To generate candidate plans that are biased toward regions of high geometric uncertainty and semantic information gain, we use Dijkstra's algorithm on 2D position coordinates to compute the shortest path between each observed-free cell in the occupancy grid and the robot's current state \mathbf{s}_i , where i is the MPC replanning index. From the set of all candidate plans \mathcal{P} , we randomly sample from the GMM for target

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

positions. This allows the updated set of randomly sampled trajectories, $\hat{\mathcal{P}}$, to be biased toward the frontiers and the highest scoring semantic regions. This trajectory sampling procedure is shown in lines 2 through 5 of Algo.1. While this procedure accounts for the path of the robot, viewing angles along the trajectory, $\bar{\psi}$, are determined by pointing the robot toward the closest frontier cell or GMM mean. These viewing angles are computed to be dynamically feasible from the robot’s current orientation with control limits.

After generating the candidate trajectories, we score each trajectory using (2) and select the path with the highest score for the robot to track. This planning procedure is repeated in a Model-Predictive Control (MPC) loop. As the replanning loop progresses, we additionally decay c in (2) with parameter β and replanning index i to gradually weigh the semantic score higher than the geometric score.

V. RESULTS

To demonstrate the contributions of our method, we first compare our geometric information gain method to two baseline methods: FisherRF [16] and Bayes’ Rays [29], on static image-pose datasets that are collected from the real world. This evaluation setup directly examines the effectiveness of VISTA’s geometric information metric, an essential component for successful task-driven robot exploration, compared to other baselines. We then incorporate semantic information and our proposed planning approach to implement the full pipeline in hardware on a quadrotor platform. On this hardware platform, we compare our method to two baselines; one using only semantic relevance and the other only geometric information gain. The semantic information-only baseline plans greedily toward the highest semantic information point in the 3DGS map, while the geometric information gain baseline is our reimplementation of RT-Guide [30]. This experiment evaluates the advantages of fusing semantic information and geometric uncertainty in robot exploration problems. Lastly, we demonstrate our full pipeline in hardware on a Boston Dynamics Spot quadruped robot to show versatility of our method to different hardware platforms. We show third-person views of the Quadrotor and Spot robot in their respective testing environments in Fig. 5.

A. Next Best View Selection Baseline Comparisons

To evaluate our geometric information gain metric, we compare against baseline approaches FisherRF [16] and Bayes’ Rays [29]. In our baseline comparisons, we train a radiance field using a predetermined set of training views for a fixed number of iterations (1000). We then apply each geometric information gain metric to select a fixed number of additional views. After including these additional views, we train the models for 1000 iterations and render images from fixed test viewpoints. We evaluate each method using the standard metrics: Peak-Signal-Noise-Ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS), and Structural Similarity Index Measure (SSIM).

We evaluate each method across six scenes: three benchmark scenes in Nerfstudio (*Plane*, *Kitchen*, and *Poster*) and

three additional datasets (*Flight*, *Clutter*, and *Adirondacks*), shown in Fig. 3. We provide the performance results of each method in Fig. 3. We find that VISTA achieves the highest PSNR and SSIM scores and the lowest LPIPS score across all scenes. Moreover, VISTA requires the fewest number of training iterations to reach the best PSNR, SSIM, and LPIPS scores in many scenes, demonstrating the VISTA’s superiority in selecting informative views compared to the other methods. For example, the best-competing method, FisherRF, requires almost twice as many training iterations to achieve the same photometric scores as VISTA, in the *Poster* and *Adirondacks* scenes. Meanwhile, VISTA requires about the same or lower computation time as the best-competing methods. These results indicate that VISTA’s information-gain metric accurately captures the information content of candidate views, compared to prior work.

B. Hardware Experiments

We first demonstrate VISTA on a custom-built quadrotor that uses a ZED-Mini stereo camera for images, and a Jetson Orin Nano onboard computer. For pose feedback, we use an OptiTrack external motion capture system, and all 3DGS training and planning is done on a desktop computer with a RTX 4090 GPU. The quadrotor receives waypoints from the offboard computer and uses an onboard state machine and PX4 flight controller for lower level control.

We compare VISTA against two different baseline approaches. The first baseline solely uses a geometric information gain to quantify trajectories that are sampled uniformly throughout the environment, and is based off the work in RT-Guide [30]. Specifically, we compute the change in the means of the Gaussians in the scene, and use this signal to find the regions of the environment where the Gaussians are changing the most. Instead of using the planning method that is used in the original paper, we adapt our method to seed the GMM with these high changing Gaussians. When trajectories are sampled from the GMM and scored, they are scored by the counts of the high uncertainty and low uncertainty Gaussians visible from each view point along the sampled trajectory. In the second baseline, the method plans toward the point from the 3DGS point cloud with the highest semantic similarity and points its heading along the velocity vector of the path.

We compare VISTA against these two baselines using three different query objects, with two different map configurations, illustrated in Fig. 5. In the first map, the query objects are not occluded in the environment. In the second map configuration, we intentionally occlude the objects from the drone. We expect most methods to be able to succeed in the first map, as the query object should be relatively easy to find and have the stopping condition trigger. In the second map, we expect methods that do not account for geometric information gain to struggle to find the query object.

We evaluate all methods on success rate (SR), time to reach (TTR), and success weighted by inverse path length (SPL), as done in [37], [38]. We enforce a maximum amount of time for the quadrotor to find the query object based on battery life. Each method is tested on a query and map for two trials,

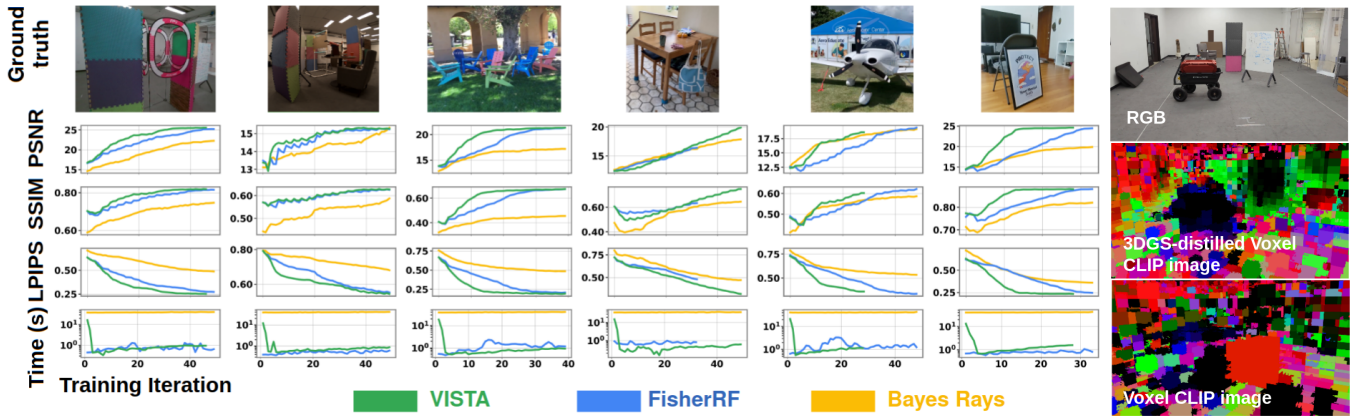


Fig. 3. Our geometric information gain metric significantly outperforms baselines FisherRF and Bayes Rays in the next best view selection task for $\sim 50K$ iterations in three visual reconstruction metrics and computation time. Six real scenes were used in this comparison, three from the Nerfstudio dataset and three additional datasets.

totaling 12 trials for each method, six on each map, with an initialization phase, where the robot turns in a small circle about its starting pose.

From the results in Table V-B, we find that all methods have some successes on the easy low-occlusion map domain. Our method has the highest success rate on this map with an 83.33% success rate over the RT-Guide baseline success rate of 66.67%, and semantic baseline success rate of 50%. In the more challenging map, we find that our method has a significant improvement over the baseline methods, where VISTA has a 100% success rate while both baselines each have a 16.67% success rate. The results suggest that VISTA is able to outperform both baselines on both maps because we reason about both semantic and geometric information gain. In these experiments, we use a voxel grid resolution of $100 \times 100 \times 50$ with a cell size of $0.2m \times 0.2m \times 0.16m$. For hyper-parameters γ , c , and β , we use 0.8, 1, and 0.9, respectively. For larger areas, β should be tuned to a lower value, as the robot will likely need to explore further before prioritizing semantic information. Additionally, the maximum trajectory steps used is $K_{max} = 50$, and the maximum number of trajectories sampled from the GMM is 10. Our voxel grid map acquires information from the 3DGS at ~ 0.28 Hz, asynchronously to the planning pipeline. The replanning step can be run up to 1 Hz (Algorithm 1) with the parameters used, however new plans are sent every 10 seconds in order to allow for the robot to sufficiently make progress along each sent trajectory.

For our second hardware platform, we use a Boston Dynamics Spot quadruped robot fitted with RGB-D cameras and onboard odometry. Only the front two cameras on the Spot robot body are used to train the 3DGS map. The offboard computer is equipped with a RTX 4090 GPU. We communicate with the Boston Dynamics Spot via the SDK and execute the task-aware plans using desired waypoint control. Qualitative results of our method on the Spot robot are shown in Fig. 5, where the semantic queries are “cone,” and “chair.” As with the quadrotor platform, Spot performs an initialization phase, turning in a small circle about the starting pose with the query object intentionally occluded from these initial views. The

same voxel grid resolution as quadrotor experiments is used, with a cell size of $0.12m \times 0.12m \times 0.02m$ is used. All other algorithmic hyper-parameters used in the Spot experiments are the same as those used in the quadrotor experiments.

C. FiGS Simulated Experiments

Further, we test our platform on a more realistic and cluttered environment of a lab space. We use the FiGS (Flying in Gaussian Splats) pipeline from [49] to generate a simulation environment using a ground truth pre-trained 3DGS (see Figure 6). Using this environment, we simulate the flight of a quadrotor and camera images using the ground truth 3DGS. In these experiments, we extend our planning approach to leverage the vertical dimension, demonstrating that our approach is not limited to planar trajectories. In Table V-B, we compare our method against the two baselines, with three different query objects and find that our method is able to achieve a higher success rate of 83.33% across all objects, while the other methods can only succeed with one of the three query objects tested at 33.33%. In these experiments, we use a voxel grid resolution of $100 \times 200 \times 30$ with a cell size of $0.12m \times 0.15m \times 0.08m$. Hyper-parameters were all the same as quadrotor and Spot experiments, with the exception of $K_{max} = 25$, and a maximum number of sampled trajectories of 5. Because there was no battery life limitation for these experiments, we used a threshold of 200 seconds for the robot to find the query object.

VI. CONCLUSION

In this work, we present an information gain metric combining both geometric information and semantic gain, and demonstrate how these metrics can be used on a real-time hardware platform to simultaneously map an environment and find an object in the map specified through natural language. We find that our method, VISTA, is fast enough to run real-time on robot hardware supported by an offboard GPU workstation, and show that by using a combined metric for semantic and geometric information gain, we can more quickly focus on areas of the map that have higher relevance to the search task. We compared our geometric information gain

Fig. 4. Our 3DGS method vs. directly projecting CLIP information.

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

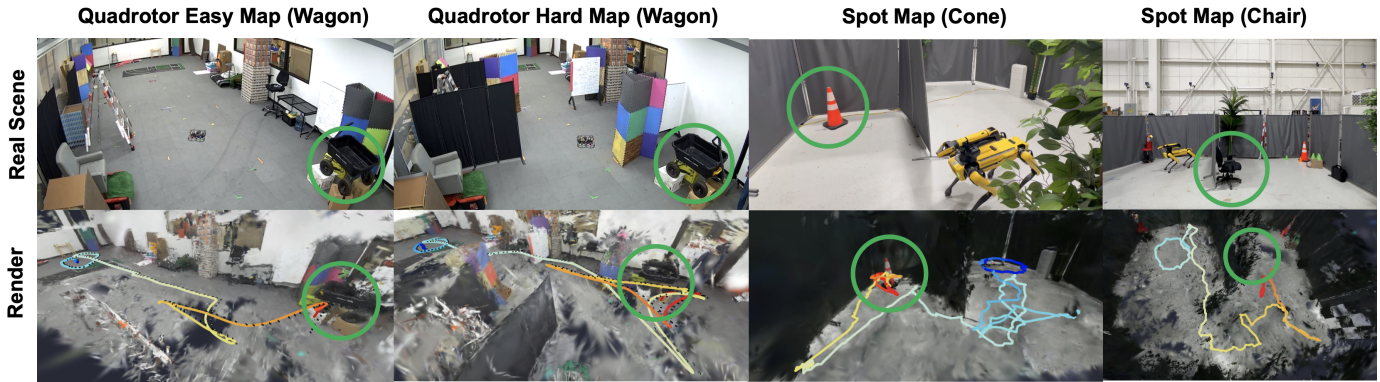


Fig. 5. The top row shows our three environments and two robots, with the search object in a green circle. The second row shows an example trajectory of the robot as it executes VISTA. Trajectories are color coded from blue (beginning) to red (end). In the first two maps (columns 1 and 2), we prompt the quadrotor with the search term “wagon.” In the last map (column 3 and 4), we prompt the spot quadruped with the terms “cone” and “chair.”

Map	Methods	Ladder (close)			Sofa (medium)			Wagon (far)		
		SR % \uparrow	TTR \downarrow	SPL % \uparrow	SR % \uparrow	TTR \downarrow	SPL % \uparrow	SR % \uparrow	TTR \downarrow	SPL % \uparrow
Easy	RT-GuIDE [30]	50	154.32	7.19	100	85.22	46.31	50	57.20	33.52
	Semantic	100	74.23	31.19	50	123.18	12.81	0	N/A	0
	VISTA [ours]	100	83.72	29.04	100	72.61	38.42	50	56.21	31.27
Hard	RT-GuIDE [30]	50	145.25	10.59	0	N/A	0	0	N/A	0
	Semantic	50	159.21	8.97	0	N/A	0	0	N/A	0
	VISTA [ours]	100	141.69	16.26	100	123.24	38.92	100	109.89	33.82
		Recycling Bin			Child Play Square			Windows		
FiGS	RT-GuIDE [30]	100	55.24	48.03	0	N/A	0	0	N/A	0
	Semantic	0	N/A	0	100	108.19	44.88	0	N/A	0
	VISTA [ours]	100	127.25	19.67	100	111.24	11.46	50	59.26	26.32

TABLE I

COMPARISON METRICS OF TIME TO TASK COMPLETION BETWEEN OUR METHOD AND BASELINE METHODS IN THE LOW-OCCCLUSION MAP (EASY MAP DOMAIN), HIGH-OCCCLUSION MAP (HARD MAP DOMAIN), AND FIGS SIMULATION LAB WORKSPACE



Fig. 6. Cluttered lab space FiGS Environment for 3D simulated quadrotor evaluations. Start position is denoted by red cross mark, and semantic queries are identified by green. The recycling bin is not visible from figure but its location denoted with a green arrow.

metric to previously published baseline methods using pre-collected datasets of images and poses, and demonstrated VISTA on two different hardware platforms in exploration tasks with varying map difficulty.

Limitations. VISTA uses CLIP features, which are known to mostly encode object-centric semantics. This limits VISTA to performing object search tasks. VISTA cannot disambiguate between multiple instances of the same object. More sophisticated VLM embeddings could enable more nuanced search tasks (avoiding danger, using landmark hints, more dynamic tasks like target following). VISTA currently requires offboard GPU compute, limiting its range and potential for field operations. VISTA also requires that the robot has its own low-level control and localization stack. In our experiments,

this is accomplished with a motion capture system for the quadrotor, and onboard SLAM system for the spot quadruped, instead of SLAM methods based on 3DGS.

REFERENCES

- [1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 8748–8763.
- [2] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.
- [3] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman *et al.*, “Foundation models in robotics: Applications, challenges, and the future,” *The International Journal of Robotics Research*, p. 02783649241281508, 2023.
- [4] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa *et al.*, “Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5021–5028.
- [5] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.
- [6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [7] B. 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*. IEEE, 1997, pp. 146–151.

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

- [8] J. Yu, H. Shen, J. Xu, and T. Zhang, "Echo: An efficient heuristic view-point determination method on frontier-based autonomous exploration for quadrotors," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5047–5054, 2023.
- [9] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, "Rapid exploration with multi-rotors: A frontier selection method for high speed flight," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2135–2142.
- [10] B. Zhou, H. Xu, and S. Shen, "Racer: Rapid collaborative exploration with a decentralized multi-uav system," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1816–1835, 2023.
- [11] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3d exploration and surface inspection," *Autonomous Robots*, vol. 42, pp. 291–306, 2018.
- [12] K. Saulnier, N. Atanasov, G. J. Pappas, and V. Kumar, "Information theoretic active exploration in signed distance fields," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4080–4085.
- [13] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "Graph-based path planning for autonomous robotic exploration in subterranean environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3105–3112.
- [14] C. Cao, H. Zhu, H. Choset, and J. Zhang, "Tare: A hierarchical framework for efficiently exploring complex 3d environments," in *Robotics: Science and Systems*, vol. 5, 2021, p. 2.
- [15] Y. Tao, Y. Wu, B. Li, F. Cladera, A. Zhou, D. Thakur, and V. Kumar, "Seer: Safe efficient exploration for aerial robots using learning to predict information gain," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1235–1241.
- [16] W. Jiang, B. Lei, and K. Daniilidis, "Fisherrf: Active view selection and mapping with radiance fields using fisher information," in *European Conference on Computer Vision*. Springer, 2024, pp. 422–440.
- [17] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 3437–3444.
- [18] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "Gslam: Dense visual slam with 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 595–19 604.
- [19] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian splatting slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 039–18 048.
- [20] H. Zhan, J. Zheng, Y. Xu, I. Reid, and H. Rezatoufghi, "Activermap: Radiance field for active mapping and planning," *arXiv preprint arXiv:2211.12656*, 2022.
- [21] S. Lee, L. Chen, J. Wang, A. Liniger, S. Kumar, and F. Yu, "Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 070–12 077, 2022.
- [22] Z. Yan, H. Yang, and H. Zha, "Active neural mapping," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 981–10 992.
- [23] D. Yan, J. Liu, F. Quan, H. Chen, and M. Fu, "Active implicit object reconstruction using uncertainty-guided next-best-view optimization," *IEEE Robotics and Automation Letters*, 2023.
- [24] R. Jin, Y. Gao, Y. Wang, Y. Wu, H. Lu, C. Xu, and F. Gao, "Gsplanner: A gaussian-splatting-based planning framework for active high-fidelity reconstruction," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 11 202–11 209.
- [25] Z. Xu, R. Jin, K. Wu, Y. Zhao, Z. Zhang, J. Zhao, F. Gao, Z. Gan, and W. Ding, "Hgs-planner: Hierarchical planning framework for active scene reconstruction using 3d gaussian splatting," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 14 161–14 167.
- [26] W. Jiang, B. Lei, K. Ashton, and K. Daniilidis, "Ag-slam: Active gaussian splatting slam," *arXiv preprint arXiv:2410.17422*, 2024.
- [27] L. Jin, X. Zhong, Y. Pan, J. Behley, C. Stachniss, and M. Popenović, "Activegvs: Active scene reconstruction using gaussian splatting," *IEEE Robotics and Automation Letters*, 2025.
- [28] G. Liu, W. Jiang, B. Lei, V. Pandey, K. Daniilidis, and N. Motee, "Beyond uncertainty: Risk-aware active view acquisition for safe robot navigation and 3d scene understanding with fisherrf," *arXiv preprint arXiv:2403.11396*, 2024.
- [29] L. Goli, C. Reading, S. Sellán, A. Jacobson, and A. Tagliasacchi, "Bayes' rays: Uncertainty quantification for neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 061–20 070.
- [30] Y. Tao, D. Ong, V. Murali, I. Spasojevic, P. Chaudhari, and V. Kumar, "Rt-guide: Real-time gaussian splatting for information-driven exploration," *IEEE Robotics and Automation Letters*, vol. 10, no. 11, pp. 11 594–11 601, 2025.
- [31] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "LERF: Language embedded radiance fields," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 19 729–19 739.
- [32] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi, "Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 676–21 685.
- [33] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "Langsplat: 3d language gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 051–20 060.
- [34] O. Shorinwa, J. Sun, and M. Schwager, "Fast-splat: Fast, ambiguity-free semantics transfer in gaussian splatting," *arXiv preprint arXiv:2411.13753*, 2024.
- [35] J. Yu, T. Chen, and M. Schwager, "Hammer: Heterogeneous, multi-robot semantic gaussian splatting," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 7270–7277, 2025.
- [36] O. Shorinwa, J. Sun, M. Schwager, and A. Majumdar, "Siren: Semantic, initialization-free registration of multi-robot gaussian splatting maps," in *Proceedings of The 9th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Lim, S. Song, and H.-W. Park, Eds., vol. 305. PMLR, 27–30 Sep 2025, pp. 3804–3827. [Online]. Available: <https://proceedings.mlr.press/v305/shorinwa25a.html>
- [37] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. 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*, 2023, pp. 23 171–23 181.
- [38] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 42–48.
- [39] D. Ong, Y. Tao, V. Murali, I. Spasojevic, V. Kumar, and P. Chaudhari, "Atlas navigator: Active task-driven language-embedded gaussian splatting," 2025. [Online]. Available: <https://arxiv.org/abs/2502.20386>
- [40] S. Papatheodorou, N. Funk, D. Tzoumanikas, C. Choi, B. Xu, and S. Leutenegger, "Finding things in the unknown: Semantic object-centric exploration with an mav," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3339–3345.
- [41] S. Barbas Laina, S. Boche, S. Papatheodorou, S. Schaefer, J. Jung, and S. Leutenegger, "Findanything: Open-vocabulary and object-centric mapping for robot exploration in any environment," *arXiv e-prints*, pp. arXiv–2504, 2025.
- [42] K. Zheng, A. Paul, and S. Tellex, "A system for generalized 3d multi-object search," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1638–1644.
- [43] J. Yu, J. E. Low, K. Nagami, and M. Schwager, "Nerfbridge: Bringing real-time, online neural radiance field training to robotics," *arXiv preprint arXiv:2305.09761*, 2023.
- [44] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, "Nerfstudio: A modular framework for neural radiance field development," in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH '23, 2023.
- [45] T. Chen, A. Swann, J. Yu, O. Shorinwa, R. Murai, M. Kennedy, and M. Schwager, "A control barrier function for safe navigation with online gaussian splatting maps," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 11 758–11 765.
- [46] F. Engelmann, F. Manhardt, M. Niemeyer, K. Tateno, and F. Tombari, "Opennerf: Open set 3d neural scene segmentation with pixel-wise features and rendered novel views," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: <https://openreview.net/forum?id=SgJAojPKb3>
- [47] S. Peng, K. Genova, C. "Jiang, A. Tagliasacchi, M. Pollefeys, and T. Funkhouser, "Openscene: 3d scene understanding with open vocabularies," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 815–824.
- [48] J. Amanatides and A. Woo, "A fast voxel traversal algorithm for ray tracing," *Proceedings of EuroGraphics*, vol. 87, 08 1987.
- [49] J. Low, M. Adang, J. Yu, K. Nagami, and M. Schwager, "Sous vide: Cooking visual drone navigation policies in a gaussian splatting vacuum," *IEEE Robotics and Automation Letters*, 2025.