HIERARCHICAL SCORING WITH 3D GAUSSIAN SPLATTING FOR INSTANCE IMAGE-GOAL NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Instance Image-Goal Navigation (IIN) requires autonomous agents to identify and navigate to a target object or location depicted in a reference image captured from any viewpoint. While recent methods leverage powerful novel view synthesis (NVS) techniques, such as 3D Gaussian splatting (3DGS), they typically rely on randomly sampling multiple viewpoints or trajectories to ensure comprehensive coverage of discriminative visual cues. This approach, however, creates significant redundancy through overlapping image samples and lacks principled view selection, substantially increasing both rendering and comparison overhead. In this paper, we introduce a novel IIN framework with a hierarchical scoring paradigm called **GauScoreMap** that estimates optimal viewpoints for target matching. Our approach integrates cross-level semantic scoring, utilizing CLIP-derived relevancy fields to identify regions with high semantic similarity to the target object class, with fine-grained local geometric scoring that performs precise pose estimation within promising regions. Extensive evaluations demonstrate that our method achieves state-of-the-art performance on simulated IIN benchmarks and real-world applicability.

1 INTRODUCTION

Instance Image-Goal Navigation (IIN) is critical in embodied navigation, requiring an agent to identify and move to the object or location depicted in a target image—often captured from any viewpoint Krantz et al. (2022). This flexibility is essential in real-world scenarios where users may provide photos from arbitrary perspectives. However, viewpoint discrepancies, cluttered scenes, and occlusions complicate the alignment of target images with the agent's observations. Effective solutions must robustly align these visual representations, enabling the agent to accurately interpret and navigate to the specified object or location.

Motivated by advances in novel view synthesis (NVS) methods, such as Neural Radiance Fields (NeRF) Mildenhall et al. (2021) and 3D Gaussian splatting (3DGS) Kerbl et al. (2023), recent approaches have begun to explore more expressive, view-consistent scene representations for IIN. Methods Cui et al. (2024); Wang et al. (2024) combine NeRF rendering with a topological graph, embedding RGB observations and learned image features into graph nodes. While this strategy retains more detailed appearance information, discretizing the environment into nodes constrains the agent's ability to observe scenes from diverse angles or navigate more complex layouts, thereby limiting truly free-view navigation.

Alternatively, 3DGS-based approaches Lei et al. (2025); Meng et al. (2024); Honda et al. (2025) preserve a continuous three-dimensional representation, offering high geometric fidelity and robust performance. However, these methods typically rely on randomly sampling multiple viewpoints Lei et al. (2025) or trajectories Meng et al. (2024); Honda et al. (2025) to ensure comprehensive coverage of discriminative visual cues. This sampling strategy in continuous 3D space creates significant redundancy through overlapping rendered images, substantially increasing both rendering and comparison overhead. The resulting trade-off between coverage and efficiency limits the practical deployment of these approaches.

To address these limitations, we introduce a novel IIN framework named **GauScoreMap** with a hierarchical scoring paradigm that efficiently estimates optimal viewpoints for target matching. Our approach eliminates excessive sampling by integrating two complementary scoring mechanisms over

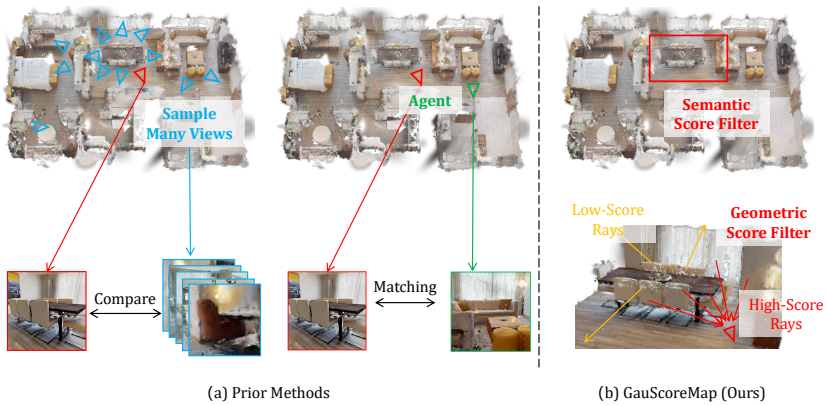


Figure 1: Overall method comparison. Prior approaches typically (i) sample many views around candidate objects or (ii) struggle to match visually dissimilar images. Our method leverages a 2-stage scoring method to semantically and geometrically locate the target image efficiently.

the built 3DGS scene. First, our global semantic scoring leverages CLIP-derived relevancy fields to identify regions with high semantic similarity to the target object class. By computing cosine similarity between CLIP text embeddings of the detected object class and features of each Gaussian, followed by thresholding and diffusion, we form coherent candidate regions containing potential target objects. Second, our local geometric scoring employs a two-stage approach: initially performing region-level scoring by comparing sampled rays from candidate regions with the goal image's DINOv2 features through cross-attention, then conducting precise pose estimation within the most promising region.

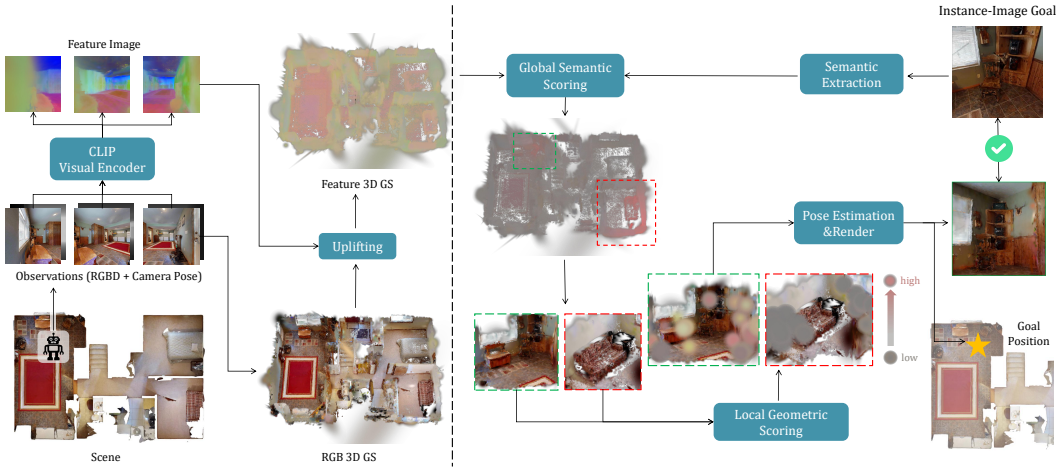
Recent advances in Gaussian splatting have enabled embodied agents to represent and explore environments with high visual fidelity. However, existing approaches often suffer from inefficiencies in view selection or struggle when appearance differ very much between observations and target images Lei et al. (2025); Meng et al. (2024) as shown in Figure 1. These limitations hinder their ability to generalize and maintain efficiency across diverse scenarios. To address these challenges, we propose a new framework that integrates global semantic reasoning with local geometric cues, enabling robust and efficient localization of target objects. By combining these complementary perspectives, our approach significantly reduces computational demands while maintaining strong accuracy, leading to state-of-the-art results in both simulated and real-world benchmarks.

The contributions of our method are mainly summarized as follows: 1) We introduce a two-tier scoring approach that combines high-level semantic alignment with fine-grained geometric matching, producing a continuous relevance map that highlights where the target image content is most likely to appear in the 3D environment. 2) We leverage the score map to identify and select the most informative viewpoints for matching, thereby obviating the need for exhaustive or random sampling throughout the environment. 3) We achieve new state-of-the-art results on instance-specific image-goal navigation benchmark data and further demonstrate our method's reliable operation in real-world indoor environments.

2 Related Work

2.1 Instance Image Goal Navigation

Deep reinforcement learning approaches have emerged as a major solution to IIN, where end-to-end policies are learnt to align current observations with target images, achieving promising simulator results through extensive training Lei et al. (2024); Qin et al. (2025). However, these reactive methods struggle to retain knowledge of explored areas in complex scenes Krantz et al. (2023), lacking explicit environment representations and degrading when agents must re-localize after losing sight of key features. To improve context retention and adaptability, map-based IIN methods incorporate spatial representations to guide navigation Yu et al. (2023); Majumdar et al. (2022); Yuan et al.



(a) Gaussian Reconstruction and Feature Lifting (b) Instance-Image Goal Localization

Figure 2: Overview of our GauScoreMap approach for Instance Image-goal Navigation. Our method consists of two main stages: (a) Gaussian Reconstruction and Feature Lifting, where we build a 3D Gaussian representation of the environment and lift CLIP features into this representation; and (b) Instance-Image Goal Localization, which uses a two-step scoring process to first identify semantically relevant regions and then precisely locate the target object instance.

(2024). Early approaches used metric maps—typically 2D bird's-eye view grids from SLAM—to track spatial locations relative to obstacles and landmarks Chaplot et al. (2020); Lei et al. (2024). While addressing some reactive strategy limitations, these 2D representations discard 3D geometry and texture details crucial for goal image matching. Recent work explores structured maps and novel view synthesis Cui et al. (2024); Wang et al. (2024); Lei et al. (2025); Meng et al. (2024); Honda et al. (2025), combining visual features with graph nodes to preserve rich appearance cues. These developments highlight that robust IIN requires representations bridging environment structure with high-resolution visual information for accurate target localization.

2.2 Novel View Synthesis in Embodied Visual Navigation

Early efforts like e2e-NeRF-nav Liu et al. (2024b) integrate online Neural Radiance Fields into the control loop for end-to-end training, but continual NeRF Mildenhall et al. (2021) updates are computationally expensive. HNR-VLN Wang et al. (2024) shifts complexity from policy learning to look-ahead synthesis by using NeRF to render candidate viewpoints for graph search. Frontier-enhanced Topological Memory Cui et al. (2024) extends this by adding "ghost" nodes to topological graphs, combining geometric reachability with appearance-based reasoning. However, discrete node representations limit diverse viewpoint observation and hinder navigation in complex layouts. GaussNav Lei et al. (2025) instead uses 3D Gaussian Splatting Kerbl et al. (2023) to preserve high-fidelity geometry and textures, while BEINGS Meng et al. (2024) employs Monte Carlo model-predictive control with hypothetical rollout rendering. Despite their effectiveness, 3DGS-based methods suffer from high computational overhead, motivating our development of a method that leverages fine-grained local visual information without extensive trajectory or viewpoint sampling.

3 Method

3.1 Overview

In Instance Image-goal Navigation (IIN), an agent navigates to a specific object instance shown in a goal image I_g. Starting from an initial position and orientation, the agent receives RGB-D observations and camera poses at each timestep, selecting actions to locate the target. Success is achieved when the agent reaches the goal vicinity within a maximum action limit.

To address IIN, we propose *Gaussian Splatting Score Maps for Visual Navigation* (GauScoreMap), illustrated in Figure 2. Our method operates in two stages: First, the agent explores the environment to build a Gaussian splatting representation and lifts 2D visual features into a 3D feature-rich Gaussian field. Second, we perform hierarchical scoring—extracting semantic information from the goal image to generate a global score map identifying candidate regions, then computing local similarity scores within these regions for precise target localization.

3.2 Gaussian Reconstruction and Feature Lifting

3.2.1 Gaussian Reconstruction

When placed in a new environment, the agent employs a frontier-based exploration strategy Yamauchi (1997); Holz et al. (2010); Juliá et al. (2012) to systematically cover the environment and collect observations for Gaussian reconstruction.

From the collected observations $\{(I_i, D_i, P_i)|i \in [0, N]\}$ (RGB images, depth maps, and camera poses), we reconstruct the RGB Gaussian splatting field using a hierarchical approach Yugay et al. (2023; 2024). For each observation subset $\{(I_i, D_i, P_i)|i \in [m, n]\}$, we initialize a submap by backprojecting the first frame's RGB image into 3D space using depth and pose. Subsequent frames densify the submap with additional Gaussian primitives. Finally, local submaps are merged into a global field.

During training, we render both color and depth from the Gaussian field. For pixel p viewing from direction v, the color and depth values are computed from the ordered set $\mathcal{S}_{v,p}$ of Gaussis:

$$\hat{I}_v(p) = \sum_{i \in \mathcal{S}_{v,p}} c_i(v) w_i(v,p), \quad \hat{D}_i(p) = \sum_{i \in \mathcal{S}_{v,p}} d_i(v) w_i(v,p) \tag{1}$$

where $\hat{I}_i(p)$ and $\hat{D}_i(p)$ are rendered color and depth values, $c_i(v)$ and $d_i(v)$ are color and depth values of the i-th Gaussian along the ray through pixel p viewing from direction v, $w_i(v,p) = \alpha_i(v,p) \prod_{j \in \mathcal{S}_{v,p}, j<i} (1 - \alpha_j(v,p))$ is the rendering weight and $\alpha_i(v,p)$ is the transparency value.

The optimization uses a combined loss:

$$\mathcal{L} = \lambda_{\text{color}} \mathcal{L}_{\text{color}} + \lambda_{\text{depth}} \mathcal{L}_{\text{depth}} \tag{2}$$

where $\mathcal{L}_{\text{color}} = \|I_i - \hat{I}_i\|_1$ and $\mathcal{L}_{\text{depth}} = \|D_i - \hat{D}_i\|_1$ are L1 losses between ground truth and rendered images/depths, and λ terms balance the loss components.

3.2.2 Feature Lifting

The visual features produced by the CLIP Radford et al. (2021) visual encoder are uplifted with simple aggregation from all collected frames Marrie et al. (2024). For each 3D Gaussian in the scene, we construct its feature representation as a weighted average of 2D features from all frames. The feature f_i of Gaussian i is:

$$f_i = \sum_{(v,p) \in S_i} \bar{w}_i(v,p) F_{v,p} \text{ with } \bar{w}_i(v,p) = \frac{w_i(v,p)}{\sum_{(v,p) \in S_i} w_i(v,p)} \tag{3}$$

where $S_i = \{(v,p) : i \in \mathcal{S}_{v,p}\}$ is the set of view-pixel pairs passing through Gaussian i. This weighting is intuitive: larger rendering weights indicate closer proximity to ray termination, so corresponding features $F_{v,p}$ contribute more significantly to Gaussian i's representation.

3.3 Global Semantic Scoring

After constructing the CLIP feature Gaussian field, we generate a global relevancy score map from the visual input. Since images contain more than just the target object, directly computing relevancy with the feature field produces noisy segmentation. We therefore use Mask-RCNN He et al. (2017) to extract the class label, which is fed to the CLIP text encoder to obtain text embedding E_T. For each Gaussian g with CLIP feature f_g, we compute a relevancy score S_g using cosine similarity:

$$S_g = \frac{E_T \cdot f_g}{\|E_T\| \cdot \|f_g\|} \tag{4}$$

This assigns a relevancy score to each Gaussian, creating a continuous score field. Applying a threshold τ yields segmented regions of Gaussians likely belonging to the target category. To address fragmentation from naive thresholding, we follow LUDVIG Marrie et al. (2024) by incorporating scene geometry and diffusing the segmentation based on feature similarity between neighboring Gaussians. This connects fragmented parts of the same instance into well-formed connected components, each representing a candidate region containing a potential target. These regions significantly reduce the search space for subsequent image-based 6D pose estimation, where we match the goal image against each candidate to precisely locate the target instance.

3.4 Local Geometric Scoring

After identifying multiple candidate regions through global semantic scoring, we need to further refine our search in two stages: first determining the most likely local region containing the target object, and then estimating the precise 6D pose within that region.

3.4.1 Local scoring for region selection

For each candidate region identified in the global scoring stage, we sample random Gaussians and generate rays in the hemisphere defined by the surface normal of each Gaussian (estimated using neighboring Gaussians). This process yields a set of ray inputs: $\{(o_i, d_i, c_i)|i \in [0, K]\}$, where o_i is the ray origin, d_i is the ray direction, and c_i is the 1-st order spherical harmonic coefficient representing the color of the ray.

Following the approach in Matteo et al. (2024), we encode these rays using a learned MLP with positional encoding as $r_i = \text{MLP}(\gamma(o_i), \gamma(d_i), \gamma(c_i))$, where $\gamma(\cdot)$ denotes the positional encoding function. This transforms the ray set into a feature representation of shape (K, C_1).

Concurrently, we process the goal image I_g through a DINOv2 Oquab et al. (2023) visual encoder to get its visual feature F_g of shape (l, C_2), where $l = h \times w$ represents the spatial dimensions of the feature map. These features are then compared through a cross-attention mechanism:

$$A = \text{CrossAttention}(r, F_g) \in \mathbb{R}^{K \times l} \tag{5}$$

By summing along the second dimension L, we get the attention score of ray k as $\hat{s}_k = \sum_{l=1}^{L} \hat{A}_{k,l}$. The ray MLP and the cross-attention module are optimized concurrently during training by minimizing the difference between predicted ray scores $\hat{s}$ and geometric ground truth ray scores s. To compute ground truth ray scores s, we leverage the insight that relevant rays should intersect at a common 3D point, indicating the position of the target image. Specifically, among the randomly sampled K rays, the most relevant rays share a key geometric property: the distance h between the ground truth camera center and its projection onto the ray should be minimal. Thus, we compute $h = ||(v_o + \ell v_d) - \mathbf{O}||_2$, where v_o and v_d are the ray origin and direction respectively, $\mathbf{O}$ is the ground truth camera center, and $\ell = \max((\mathbf{O} - v_o) \cdot v_d, , 0)$ is the projection length of vector $(\mathbf{O} - v_o)$ onto the ray. The distance h ranges from 0 to $+\infty$, with $h = 0$ indicating the ray passes exactly through the camera center. Finally, we map the distance h_k of the k-th ray to its ray score using:

$$\delta_k = 1 - tanh(h_k), \quad s_k = \delta_k \frac{L}{\sum_{k=1}^{K} \delta} \tag{6}$$

Then a softmax function is required to normalize all ground truth ray scores as $\{s_k|k \in [1, K]\} = softmax(\{s_k|k \in [1, K]\})$. Then we train the MLP and cross attention modules by minimizing the $L2$ loss $\mathcal{L} = \frac{1}{K} \sum_{k=1}^{K} ||s_k - \hat{s}_k||$ between the predicted ray scores $\hat{S}$ and the ground truth ray scores S.

Table 1: Success rate and SPL comparison of our method with four sets of baseline methods.

Category	Method	SR ↑	SPL ↑
MultiON Transfer	MultiON Baseline Wani et al. (2020)	0.066	0.045
	MultiON Implicit Marza et al. (2023)	0.143	0.107
	MultiON Camera Chen et al. (2022)	0.186	0.142
SOTA IIN	Mod-IIN Krantz et al. (2023)	0.561	0.233
	IEVE Mask RCNN Lei et al. (2024)	0.684	0.241
	IEVE InternImage Lei et al. (2024)	0.702	0.252
SOTA IIN with Scene/3DGS Map	Mod-IIN (Scene Map) Krantz et al. (2023)	0.563	0.323
	IEVE Mask RCNN (Scene Map) Lei et al. (2024)	0.683	0.331
	IEVE InternImage (Scene Map) Lei et al. (2024)	0.705	0.347
	GaussNav (3DGS Map) Lei et al. (2025)	0.725	0.578
	GauScoreMap (3DGS Map)	**0.784**	**0.605**

3.4.2 Fine-Grained Pose Estimation for Precise Localization

Once we've identified the most promising region, we perform a second, more dense sampling of Gaussian-ray pairs within this region. This denser sampling allows for more precise localization of the target object. We select top k Gaussian-ray pairs with the highest scores from the new samples, and perform triangulation as described in Matteo et al. (2024) to estimate the 6D pose (position and orientation) of the target object. This two-stage scoring approach—first at the region level and then at the pose level—enables our system to efficiently narrow down the search space before performing precise localization, significantly improving both the efficiency and accuracy of the object localization process. These two scoring steps use the same pretrained ray-image cross attention neural network by minimizing the difference between the predicted camera 6D pose and the gt camera 6D pose as Matteo et al. (2024).

4 Experiment

4.1 Experiment Setup

Dataset. We conduct our experiments using the Habitat Szot et al. (2021) simulator. For scene data, we utilize the Habitat-Matterport 3D dataset (HM3D) Yadav et al. (2023). Specifically, we use version 0.2 of the HM3D dataset and follow the Instance ImageGoal Navigation (IIN) Krantz et al. (2022) in the Habitat Navigation Challenge 2023[1]. We evaluate our method on the 1,000 validation episodes specified by Krantz *et al.* Krantz et al. (2022). This validation subset encompasses six object categories: {*chair, couch, bed, toilet, television, plant*} and includes 795 unique object instances.

Agent Configuration. We adopt the standard agent configuration from the Habitat Navigation Challenge 2023. The agent is modeled as a rigid-body cylinder with zero turning radius, standing 1.41m tall with a radius of 0.17m. A forward-facing RGB-D camera is mounted at a height of 1.31m. At each time step t, the agent receives observations consisting of RGB images, depth maps, and sensor poses. The agent operates in a continuous action space with four dimensions: *linear velocity, angular velocity, camera pitch velocity, and velocity stop*. Each action dimension accepts values between -1 and 1, which are then scaled according to their respective configuration parameters. The maximum linear speed is $35cm/frame$, while the maximum angular velocity is $60°/frame$.

Evaluation Metrics. Our evaluation incorporates both effectiveness and navigation efficiency metrics. The primary metrics we use are SR (Success Rate) and SPL (Success weighted by Path Length). A navigation attempt is considered successful when the agent executes the stop action within a 1.0m radius of the target object and can visually detect the object by adjusting its camera orientation. The SPL metric, as introduced by Anderson et al. Anderson et al. (2018), provides a balanced assessment of navigation efficiency by considering both success and path optimality.

[1] https://aihabitat.org/challenge/2023/

Table 2: Ablation study of our method.

Method	SR ↑	SPL ↑
GauScoreMap	0.784	0.605
GauScoreMap w.o. Global Semantic Scoring	0.608	0.419
GauScoreMap w.o. Local Geometric Scoring	0.421	0.310
GauScoreMap w. GT Match	0.842	0.650
GauScoreMap w. GT Global Localization	0.944	0.742

4.2 Comparison with State-of-the-art Methods

We compare our method against a comprehensive set of baseline approaches as presented in Table 1, with baseline results sourced from GaussNav Lei et al. (2025). The comparison methods are organized into three categories:

MultiON Transfer Methods. These approaches were originally designed for the MultiON task, which shares similarities with scene-specific map representations: (1) MultiON Baseline, a standard implementation of Wani et al. (2020); (2) MultiON Implicit Marza et al. (2023), which learns an implicit neural representation; and (3) MultiON Camera Chen et al. (2022), which develops an active camera movement policy. These methods receive the semantic category of the target object as input.

State-of-the-art IIN Methods. Leading IIN approaches include: (1) Mod-IIN Krantz et al. (2022), which decomposes the task into exploration, goal instance re-identification, goal localization, and local navigation; (2) IEVE Mask RCNN Lei et al. (2024), which implements a modular architecture using Mask RCNN He et al. (2017) for object detection; and (3) IEVE InternImage, an enhanced variant with a more powerful detector.

SOTA IIN with Scene/3DGS Map. This category includes the above methods when augmented with different map representations: (1-3) Mod-IIN, IEVE Mask RCNN, and IEVE InternImage with traditional scene maps; (4) GaussNav Lei et al. (2025), which utilizes 3D Gaussian Splatting maps; and (5) our proposed approach, which also leverages 3DGS maps but enhances performance.

As shown in Table 1, our method significantly outperforms all baselines, achieving the highest success rate (0.784) and SPL (0.605). Notably, our approach surpasses GaussNav by 5.9% in success rate and 2.7% in SPL. While both GaussNav Lei et al. (2025) and our method utilize Gaussian splatting fields for navigation, our method has better performance with our enhanced localization capabilities.

4.3 Ablation Study

We ablate the design choices of our method and show their inflences on the final performance in Table 2. And we analyze each module of our method:

GauScoreMap w.o. Global Semantic Scoring. The global semantic scoring module serves as a prefilter to extract candidate local regions for finer local localization. From Table 2, we can see that without global semantic scoring, with only local geometric scoring to produce the predicted target position, the success rate drops by 17.6%. This is because indoor scenes have severe occlusion and complicated spatial distributions. Simply sampling Gaussians and computing the relationships between ray features and Gaussians suffers from ambiguity and inefficiency. The global semantic scoring effectively narrows down the search space by identifying semantically relevant regions first, allowing the local geometric scoring to focus on promising areas. This two-stage approach significantly improves both accuracy and computational efficiency compared to relying solely on local visual scoring.

GauScoreMap w.o. Local Geometric Scoring. When we remove the local geometric scoring component and rely only on global semantic scoring, performance decreases dramatically with a 36.3% drop in success rate. This substantial decline highlights the critical role of fine-grained geometric matching in precisely localizing the target object. While global semantic scoring can identify candidate regions containing objects of the target category, it lacks the precision to distinguish specific object instances with similar semantic properties. The local geometric scoring module provides this

Table 3: Time Efficiency and Peak VRAM Usage of our method

Stage	Substage	Time	Peak VRAM Usage
Training	GS Recon (GaussNav Lei et al. (2025))	65 m(inutes)	11.0 GB
	GS Recon (Ours)	15 m(inutes)	3.6 GB
	Local Scoring Function Training	15 m(inutes)	5.5 GB
Inference	Semantic Extraction (Mask-RCNN forward)	0.10 s	3.5 GB
	Global Scoring (Relevancy score calculation)	1.12s	1.1 GB
	Local Scoring (Gaussian Sampling)	0.59 s	-
	Local Scoring (Candidate Selection)	0.32 s	4.2 GB
	Local Scoring (Pose Estimation)	0.17 s	4.2 GB

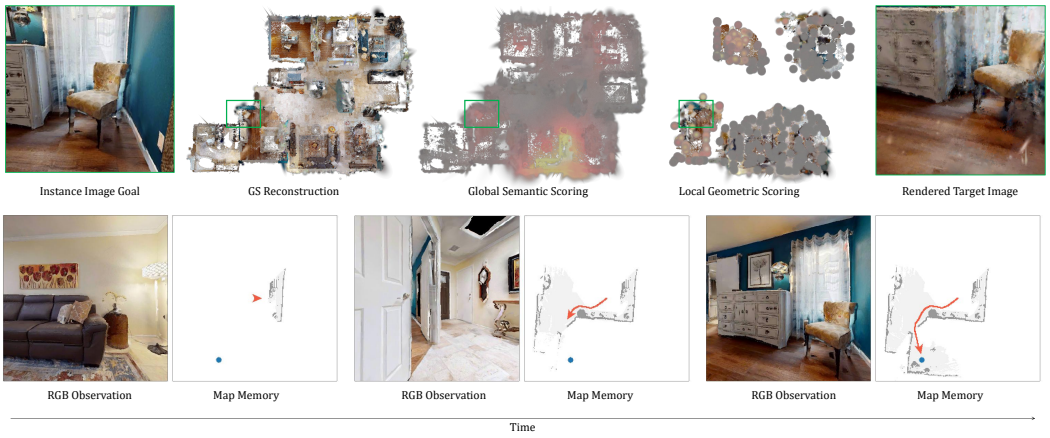


Figure 3: The localization by scoring (the first row) and the navigation process (the second row) of an episode of scene *5cdEh9F2hJL* in HM3D Yadav et al. (2023).

crucial instance-level discrimination by establishing detailed geometric correspondences between the goal image features and the 3D scene representation.

GauScoreMap w. GT Match. Using ground truth instance matching improves success rate by 5.8%, revealing room for enhancement in instance-level recognition and matching. This gap suggests that better visual feature extraction and matching could further boost performance.

GauScoreMap w. GT Global Localization. With ground truth global localization, our method achieves 94.4% success, demonstrating highly effective local navigation from accurate position estimates. This indicates that global localization is the primary bottleneck, and improving the global semantic scoring module could approach this upper bound.

4.4 Time Efficiency and Peak VRAM Usage Evaluation

Table 3 evaluates the time efficiency and peak VRAM usage of our method. We analyze each substage across training and inference phases using three HM3D Yadav et al. (2023) scenes averaging 80 m^2 with $\sim$1000 images each. In the training stage, compared to GaussNav Lei et al. (2025), our submap division strategy reduces GS reconstruction time by 50 minutes while requiring only one-third of the VRAM. For local scoring function training, we save 30 minutes over 6DGS Matteo et al. (2024) (45 minutes) by restricting the camera pose search space to navigable areas defined by the reconstructed scene.In the inference stage, all substages are efficient except global scoring, which computes relevancy for all Gaussians in the scene. We address this by sparsifying the GS scene to $\sim$600,000 Gaussians, and this preserves semantic accuracy since CLIP features remain embedded in target objects regardless of Gaussian count due to their distinctive semantic properties.

Table 4: Locating Success Rate of our method under different portions of gaussian deletion

	Delete 0%	Delete 20%	Delete 40%	Delete 60%
SR	100%	100%	90%	75%

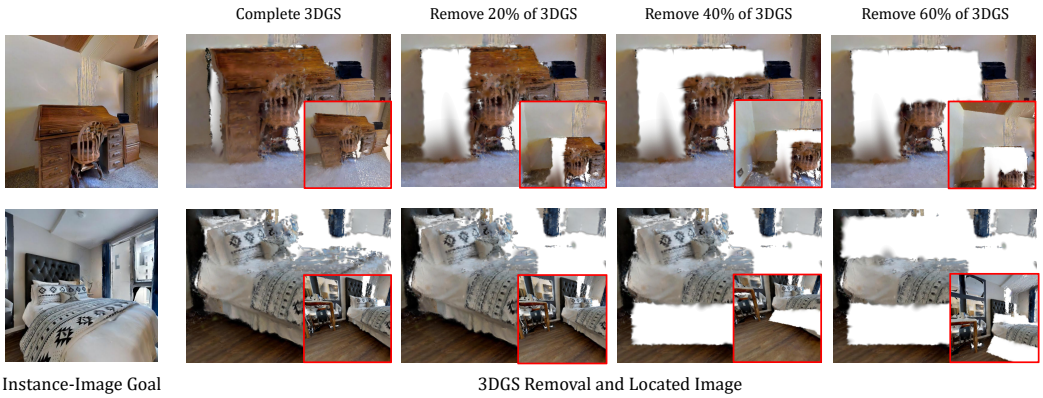


Figure 4: The robustness of our method under different portions of gaussian deletion. The red boxed image of the bottom right corner is the located image by our method.

4.5 Robustness to Incomplete Scene Exploration

Since our method relies on a reconstructed GS scene for navigation, we evaluate its robustness to incomplete exploration where target objects may be partially occluded or incompletely reconstructed. We simulate this by manually deleting portions of target object Gaussians to test localization performance under degraded conditions. We tested 20 successful episodes across 5 HM3D Yadav et al. (2023) scenes, progressively deleting around 20%, 40%, and 60% of target region Gaussians. Table 4 and Figure 4 show that our method maintains high localization success even at 40% deletion. This robustness stems from two factors: the semantic scoring accurately identifies candidate objects despite incomplete shapes, while local geometric scoring leverages surrounding texture details to determine the correct camera pose.

4.6 Visual Scoring and Navigation Results

Figure 3 illustrates the scoring and navigation results of our method. The first row demonstrates how our hierarchical scoring approach localizes the target image and renders it using reconstructed Gaussian splats. The second row shows the agent navigating to the identified position in a Habitat simulator Szot et al. (2021). Additional examples are provided in the appendix.

5 Conclusion

In this work, we introduce a novel Instance Image-Goal Navigation framework that tackles the principal challenges of viewpoint variation, semantic ambiguity, and complex scene layouts. By combining two-level semantic scoring with fine-grained geometric scoring, the method yields a continuous score map that obviates the need for exhaustive or random viewpoint sampling. Empirical evaluations on simulated benchmarks confirm state-of-the-art performance, underscoring the method's effectiveness and practical applicability. Furthermore, we deploy the proposed approach on a humanoid agent and validate its performance in real-world indoor environments. A key limitation of our method is that it focuses primarily on static environments and relies on a pre-built GS scene map. Future directions may include simultaneously exploring and locating the target image without accuracy loss.

REFERENCES

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.

Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12875–12884, 2020.

Peihao Chen, Dongyu Ji, Kunyang Lin, Weiwen Hu, Wenbing Huang, Thomas Li, Mingkui Tan, and Chuang Gan. Learning active camera for multi-object navigation. *Advances in Neural Information Processing Systems*, 35:28670–28682, 2022.

Xinru Cui, Qiming Liu, Zhe Liu, and Hesheng Wang. Frontier-enhanced topological memory with improved exploration awareness for embodied visual navigation. In *European Conference on Computer Vision*, pp. 296–313. Springer, 2024.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Dirk Holz, Nicola Basilico, Francesco Amigoni, and Sven Behnke. Evaluating the efficiency of frontier-based exploration strategies. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pp. 1–8. VDE, 2010.

Kohei Honda, Takeshi Ishita, Yasuhiro Yoshimura, and Ryo Yonetani. Gsplatvnm: Point-of-view synthesis for visual navigation models using gaussian splatting. *arXiv preprint arXiv:2503.05152*, 2025.

Miguel Juliá, Arturo Gil, and Oscar Reinoso. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots*, 33:427–444, 2012.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-specific image goal navigation: Training embodied agents to find object instances. *arXiv preprint arXiv:2211.15876*, 2022.

Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to objects specified by images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10916–10925, 2023.

Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of field robotics*, 36(2):416–446, 2019.

Xiaohan Lei, Min Wang, Wengang Zhou, Li Li, and Houqiang Li. Instance-aware exploration-verification-exploitation for instance imagegoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16329–16339, 2024.

Xiaohan Lei, Min Wang, Wengang Zhou, and Houqiang Li. Gaussnav: Gaussian splatting for visual navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Changkun Liu, Shuai Chen, Yash Bhalgat, Siyan Hu, Ming Cheng, Zirui Wang, Victor Adrian Prisacariu, and Tristan Braud. Gs-cpr: Efficient camera pose refinement via 3d gaussian splatting. *arXiv preprint arXiv:2408.11085*, 2024a.

Qiming Liu, Haoran Xin, Zhe Liu, and Hesheng Wang. Integrating neural radiance fields end-to-end for cognitive visuomotor navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024b.

Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35:32340–32352, 2022.

Juliette Marrie, Romain Ménégaux, Michael Arbel, Diane Larlus, and Julien Mairal. Ludvig: Learning-free uplifting of 2d visual features to gaussian splatting scenes. *arXiv preprint arXiv:2410.14462*, 2024.

Pierre Marza, Laetitia Matignon, Olivier Simonin, and Christian Wolf. Multi-object navigation with dynamically learned neural implicit representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11004–11015, 2023.

Bortolon Matteo, Theodore Tsesmelis, Stuart James, Fabio Poiesi, and Alessio Del Bue. 6dgs: 6d pose estimation from a single image and a 3d gaussian splatting model. In *European Conference on Computer Vision*, pp. 420–436. Springer, 2024.

Wugang Meng, Tianfu Wu, Huan Yin, and Fumin Zhang. Beings: Bayesian embodied image-goal navigation with gaussian splatting. *arXiv preprint arXiv:2409.10216*, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Yiran Qin, Ao Sun, Yuze Hong, Benyou Wang, and Ruimao Zhang. Navigatediff: Visual predictors are zero-shot navigation assistants. *arXiv preprint arXiv:2502.13894*, 2025.

Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, pp. 5. Kobe, 2009.

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*, pp. 8748–8763. PmLR, 2021.

Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3437–3444. IEEE, 2023.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8248–8258, 2022.

Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12922–12931, 2022.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, Junjie Hu, Ming Jiang, and Shuqiang Jiang. Lookahead exploration with neural radiance representation for continuous vision-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13753–13762, 2024.

Saim Wani, Shivansh Patel, Unnat Jain, Angel Chang, and Manolis Savva. Multion: Benchmarking semantic map memory using multi-object navigation. *Advances in Neural Information Processing Systems*, 33:9700–9712, 2020.

Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9068–9079, 2018.

Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-matterport 3d semantics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4927–4936, 2023.

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'*, pp. 146–151. IEEE, 1997.

Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1323–1330. IEEE, 2021.

Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3554–3560. IEEE, 2023.

Shuaihang Yuan, Hao Huang, Yu Hao, Congcong Wen, Anthony Tzes, and Yi Fang. Gamap: Zero-shot object goal navigation with multi-scale geometric-affordance guidance. volume 37, pp. 39386–39408, 2024.

Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023.

Vladimir Yugay, Theo Gevers, and Martin R Oswald. Magic-slam: Multi-agent gaussian globally consistent slam. *arXiv preprint arXiv:2411.16785*, 2024.

A Appendix

A.1 Extra Simulation Results

We present two additional examples of our method navigating to instance image goals in Figure 5 and Figure 6. Each example demonstrates both the target localization process (first row) and the navigation execution (second row).

In the first row, we illustrate the complete target localization pipeline from left to right: the instance image goal, the Gaussian Splatting (GS) reconstruction of the scene, the heat map generated during the global semantic scoring step, the local score map within segmented candidate regions, and the rendered target image using the pose estimated by the local geometric scoring step.

The second row shows three key stages of the navigation process as the agent moves toward the final goal position.

A.2 Localization of other objects

Since our method encodes CLIP features into the Gaussian field, it can localize objects beyond the six evaluation categories (*chair, couch, bed, toilet, television, plant*) in HM3D Yadav et al. (2023). Figure 7 demonstrates successful localization of *lamp*, *refrigerator*, and *bicycle* in Gibson Xia et al. (2018) and ReplicaCAD Straub et al. (2019) scenes.

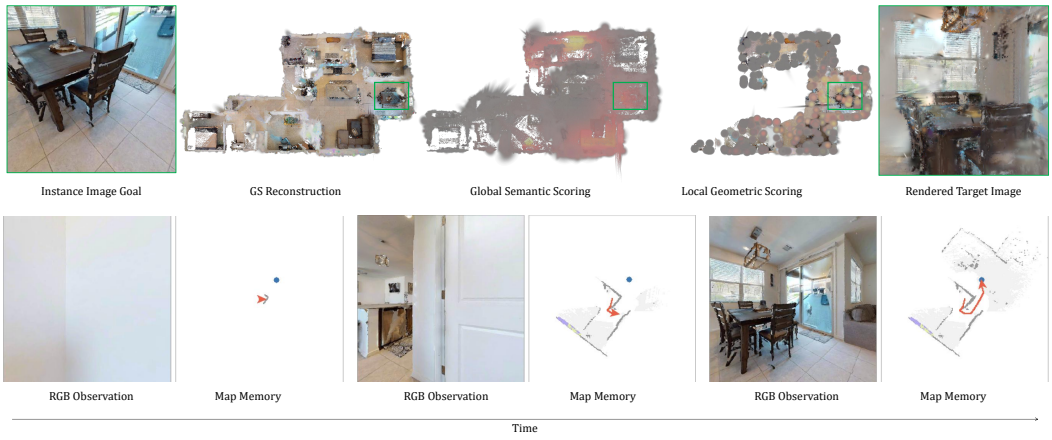


Figure 5: The target localization (the first row) and the navigation process (the second row) of an episode of scene *Nfvxx8J5NCo* in HM3D Yadav et al. (2023)'s validation set.

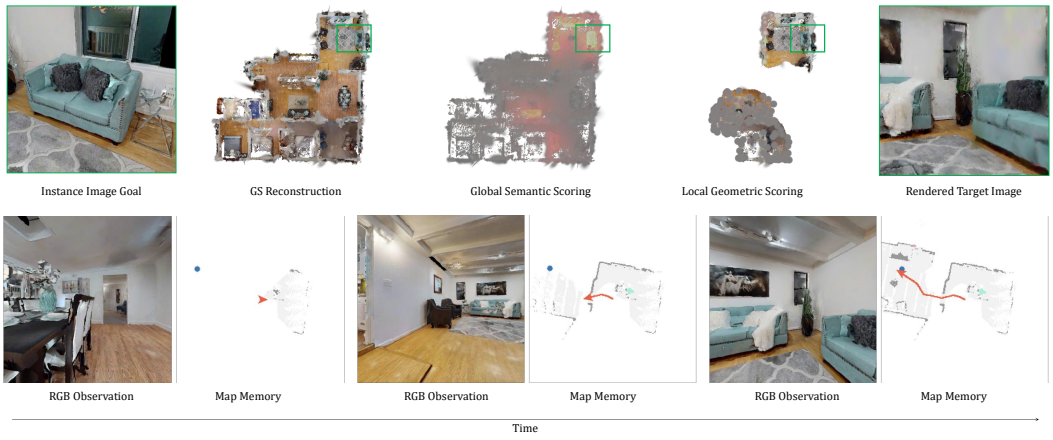


Figure 6: The target localization (the first row) and the navigation process (the second row) of an episode of scene *Dd4bFSTQ8gi* in HM3D Yadav et al. (2023)'s validation set.

A.3 Comparison with other NeRF/GS-based localization methods

Since localization is the key of our method for successful IIN navigation, we validate our localization module by replacing only this component while keeping other parts unchanged and compare against alternative methods. For NeRF-based approaches, we combine iNeRF Yen-Chen et al. (2021) for pose estimation with NeRF-SLAM Rosinol et al. (2023) for scene reconstruction. For 3DGS methods, we evaluate 6DGS Matteo et al. (2024) and GS-CPR Liu et al. (2024a), both operating on our reconstructed 3DGS maps.

We evaluate on a 100-episode subset of the HM3D validation set, with navigation success rates shown in Table 5.

iNeRF achieves 0% SR due to NeRF-SLAM's reconstruction failures: the system exhausts GPU memory when allocating large implicit volumes for HM3D apartments. This reflects well-documented limitations of implicit NeRF pipelines in large, complex indoor scenes Tancik et al. (2022); Turki et al. (2022). Gaussian splatting scales better through decomposition into smaller, independent segments.

GS-CPR also achieves 0% SR because it requires close spatial overlap between query and database images—rarely satisfied in large-scale scenarios. 6DGS attains 58% SR, respectable given the localization difficulty in complex indoor structures, but substantially below our method's performance.

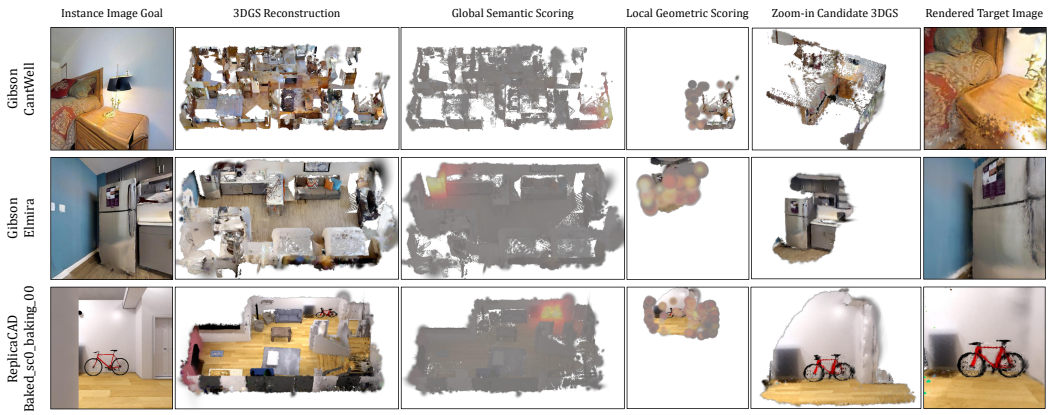


Figure 7: Localization of *Lamp* (first row), *Refridgerator* (second row) and *Bicycle* (third row) on Gibson and ReplicaCAD scenes.

Method	SR
iNeRF Yen-Chen et al. (2021)	0 (OOM)
GS-CPR Liu et al. (2024a)	0
6DGS Matteo et al. (2024)	0.58
GauScoreMap (Ours)	0.76

Table 5: Navigation SR comparison with different localization methods

These results demonstrate that our method achieves an effective balance between computational efficiency and navigation performance in the IIN task, substantially outperforming alternative localization approaches in large-scale indoor environments.

A.4 Real World Experiments

A.4.1 Environment

The layout of the testing field used for the demonstrations is illustrated in Figure 8. The environment is intentionally cluttered with numerous common household items to closely mimic real-world conditions. Additionally, several boxes are strategically placed throughout the area to introduce obstacles, thereby increasing the complexity and challenging the robot's navigation and detection capabilities.

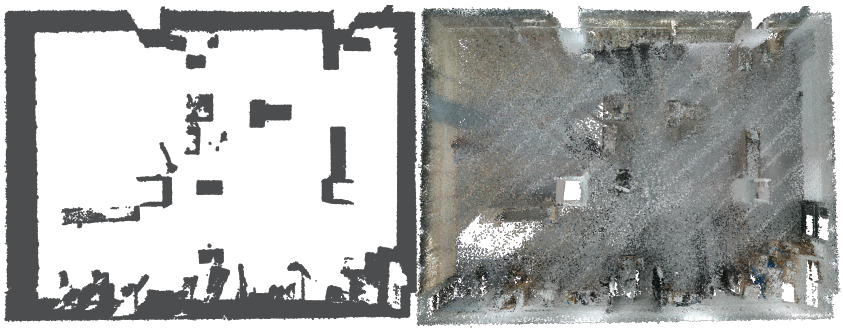


Figure 8: Layout of the testing field (left) along with point-cloud construction (right).

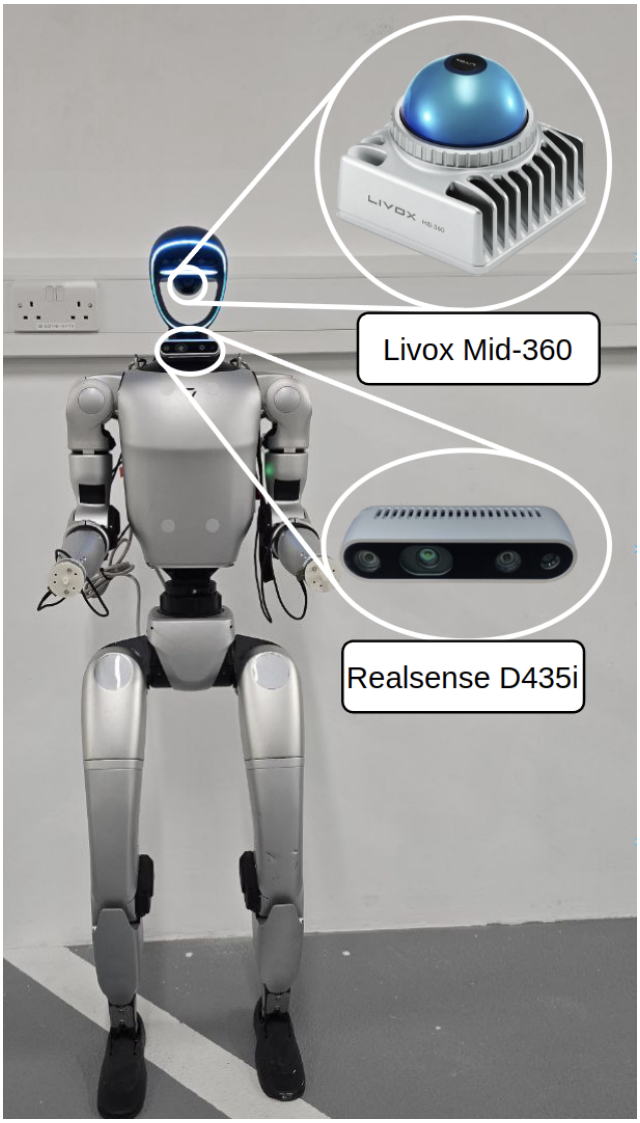


Figure 9: Unitree G1 robotic platform.

A.4.2 Robotic Platform

For this study, we utilized a Unitree G1 humanoid robot equipped with a Livox Mid-360 LiDAR and a RealSense D435i RGB-D camera. Simultaneous localization and mapping (SLAM), path planning, and low-level control are executed entirely on the robot's onboard computer. In addition to that image-goal navigation process was performed on an external laptop connected to the robot via an Ethernet cable. A photograph of the robotic platform is provided in Figure 9.

A.4.3 Odometry and Mapping

We employed RTAB-Map Labbé & Michaud (2019) as the primary module for pose estimation, mapping, and localization. The robot leverages the onboard Livox Mid-360 LiDAR to compute odometry through RTAB-Map's ICP-based odometry module. Additionally, RGB-D images captured by the RealSense camera are integrated into RTAB-Map's SLAM module, enabling robust localization, mapping, and global loop closure detection.

A.4.4 Data Collection and GS Reconstruction

To formulate the input for gaussian splatting reconstruction, for each recorded frame, the humanoid robot collect the RGB image, the 7D camera pose (a 3D camera position and a 4D camera quaternion), and the Lidar point cloud. In this real-world setting, we use the Lidar point cloud as the geometric scaffold instead of the depth map recorded by the realsense camera because the recorded depth map contains lots of artifacts and is very inaccurate in a big environment, whereas the Lidar point cloud is more reliable.

After collecting the data, we first merge the Lidar point cloud of all frames into a complete one using the collected camera poses as shown in Figure 8. This point cloud serves as the initialization of the gaussian splatting field, then we optimize the gaussian splatting field as the usual way in Kerbl et al. (2023).

A.4.5 Safe Path Planning

Safe navigation and obstacle avoidance are achieved using the ROS Navigation Stack Quigley et al. (2009). This framework integrates sensor inputs from LiDAR and RGB-D cameras to generate occupancy grid maps and perform path planning. Specifically, the navigation stack utilizes costmap-based planning algorithms such as Dijkstra's algorithm and the Dynamic Window Approach (DWA) to calculate collision-free paths in real-time, ensuring robust and safe trajectories for the humanoid robot within cluttered indoor environments.

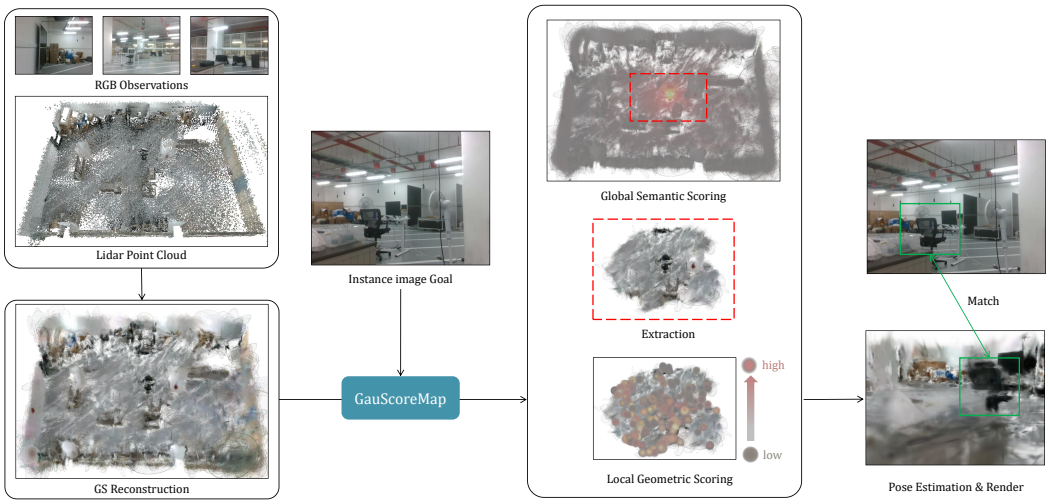


Figure 10: An example of finding the goal position of an instance image goal dipicting a chair with our collected real-world data.

A.4.6 An example of IIN using our GauScoreMap

We demonstrate an example of locating an instance image goal using our collected real-world data in Figure 10. First, we leverage the merged LiDAR point cloud and calibrated RGB images to reconstruct the Gaussian splatting field of the scene, as shown in the left portion of Figure 10.

An instance image goal depicting a chair is then provided to our GauScoreMap method, which processes it through two sequential scoring stages. The global semantic scoring stage generates a coarse localization map that roughly identifies the chair's position, while the local geometric scoring stage produces a refined location estimate. The top-k scoring rays are subsequently selected to estimate the camera pose and render an image of the target position.

As demonstrated on the right side of Figure 10, the rendered image successfully captures the chair specified in the instance image goal.