Gen-NAV: Generative Simulation with Gaussian Photorealistic Rendering for Evolving Environments

Yinghao Shuai[1*], Saining Zhang[2*†], Zijian Jiang[1], Yuantao Chen[3], Hao Zhao[4†]

***Abstract*— Generative simulation has revolutionized robotic manipulation, yet its systematic application to Vision-and-Language Navigation (VLN) remains underexplored, largely due to the lack of suitable simulation infrastructure. In this work, we propose Gen-NAV, the first generative simulation framework for VLN built upon Gaussian Splatting. Gen-NAV can dynamically generate diverse objects and environments, enabling realistic and scalable navigation scenes for training and evaluation. By combining Gaussian rendering with mesh-based geometry, Gen-NAV enables photorealistic rendering while preserving reliable geometry for navigation and collision handling. Leveraging this capability, we introduce a novel task, Lifelong Navigation in Evolving Environments, which more faithfully reflects real-world robotic navigation. In this setting, agents must sequentially navigate to multiple target objects while distractor configurations change between subtasks, resulting in a dynamic and long-horizon navigation problem. Such scenarios are difficult to define and scale in conventional simulators, but arise naturally within our generative framework. Gen-NAV distinguishes itself from existing simulators through realistic object diversity, dynamic distractor rearrangements and evolving environments across navigation subtasks. We evaluate representative state-of-the-art VLN methods, including the modular VLFM and the end-to-end OVON-DAgger, on our benchmark, and observe consistently poor performance. These results highlight the challenges posed by dynamic multi-object navigation and position Gen-NAV as a promising platform for advancing VLN research in ever-changing environments. The code will be released at `https://github.com/illusion2020/Gen-NAV`.**

I. INTRODUCTION

Vision-and-Language Navigation (VLN) requires agents to interpret natural language instructions and navigate through complex, unfamiliar environments. Developing and evaluating such agents critically depends on simulation environments that are both visually realistic and semantically rich. However, existing simulation approaches [1], [2] predominantly rely on static, low-fidelity mesh representations, which limit rendering quality and restrict environmental flexibility. Moreover, widely used VLN datasets [3]–[6] are relatively small in scale and focus on a narrow set of common household objects, such as beds, tables, and chairs. This lack of object diversity constrains the variability of navigation scenarios, hindering robust training and generalization of VLN models.

Generative simulation has recently emerged as a promising paradigm for addressing these limitations. It has been successfully applied in robotics, computer vision, and artificial intelligence [7]–[9], enabling the creation of flexible, customizable virtual environments and large-scale task-specific data. By synthesizing realistic scenes and diverse objects, generative simulation improves both visual fidelity and data diversity, which are crucial for learning robust embodied agents. Despite this progress, however, generative simulation remains largely underexplored in the context of VLN, primarily due to the lack of suitable simulation frameworks.

In this work, we introduce **Gen-NAV**, the first generative simulation framework for VLN built upon 3D Gaussian Splatting (3DGS) [10]–[13]. Unlike traditional mesh-based simulators that suffer from limited rendering quality and constrained object vocabularies, Gen-NAV enables photorealistic scene reconstruction and efficient real-time rendering through 3DGS, while supporting an open-vocabulary set of objects via an automatic 3D generation pipeline. As illustrated in Figure 1, Gen-NAV can dynamically generate diverse objects—including items such as headphones, cameras, and LEGO toys—and integrate them into evolving scenes. This capability enables more comprehensive training and evaluation of VLN algorithms in visually rich and semantically diverse environments.

Motivated by the dynamic nature of real-world navigation, we further introduce a new task: **Lifelong Navigation in Evolving Environments**. Conventional VLN tasks typically assume static goals and fixed environments, which fail to capture real-world conditions where surrounding object configurations may change over time. In contrast, our proposed task (Figure 1) requires agents to sequentially navigate toward multiple sub-goals while the environment evolves between subtasks, resulting in a long-horizon and dynamic navigation setting. Such scenarios are difficult to define and scale in traditional simulators, but arise naturally within the flexibility of generative simulation.

To support this task, Gen-NAV provides a unified framework for constructing interactive navigation environments. It seamlessly integrates a wide range of objects—including furniture, everyday items, and toys—into scenes guided by language descriptions. Furthermore, we design an object placement module that arranges generated objects in diverse yet semantically meaningful configurations, enabling dynamic object placement and rearrangement across navigation subtasks.

In summary, our contributions are threefold:

- We introduce **Gen-NAV**, a generative simulation framework based on 3D Gaussian Splatting that enables photorealistic, flexible, and dynamic environments for VLN.

* Equal contribution. [1]Tongji University; [2]Nanyang Technological University; [3]The Chinese University of Hong Kong, Shenzhen; [4]Tsinghua University.

† Corresponding author.



Fig. 1. Gen-NAV integrates 3D Gaussian Splatting for photorealistic indoor scene reconstruction and generative object synthesis to support dynamic navigation environments (top). The framework enables realistic rendering of complex scenes with diverse object instances, such as headphones, cameras, and LEGO toys, which can be dynamically generated and rearranged. The proposed **Lifelong Navigation in Evolving Environments** task (bottom) requires agents to navigate toward a sequence of object goals (green boxes), while the environment changes between subtasks through object rearrangements (red boxes). This setting evaluates an agent's ability to perform sequential navigation under evolving scene configurations.

- We propose the **Lifelong Navigation in Evolving Environments** task, a challenging benchmark in which agents must navigate toward multiple goals while adapting to ongoing environmental changes.
- We demonstrate that generative simulation in Gen-NAV supports dynamic object generation and rearrangement, creating interactive environments that better reflect the complexity of real-world navigation.

II. Related Work

A. Visual Navigation Benchmarks.

Simulation-based benchmarks are central to embodied visual navigation [14]. Existing simulators are typically built on procedurally generated environments, such as ProcTHOR [1] and HSSD [2], or scanned real-world scenes, including Replica [3], Matterport3D [4], HM3D [5], and Gibson [15]. Procedural environments offer scalability but limited realism, whereas scanned scenes provide higher visual fidelity but remain constrained in scale and diversity. These platforms support tasks such as PointNav [16], ObjectNav [17], and vision-and-language navigation. Recent ObjectNav benchmarks further consider sequential goals [18], open-vocabulary categories [19], and instance-level targets [20]. However, most assume static environments. In contrast, we introduce a dynamic multi-object navigation benchmark in which distractor configurations evolve across subtasks.

B. Object Goal Navigation Methods.

ObjectNav methods can be broadly divided into modular and end-to-end approaches. Modular methods separate perception, mapping, and planning, often using semantic maps and frontier-based or learned exploration [21]–[23]. End-to-end methods directly predict actions from observations using recurrent or transformer-based policies trained with imitation or reinforcement learning [24], [25]. Recent work also explores world-model-based planning [26] and diffusion bridge policies [27]. While modular methods benefit from explicit structure, end-to-end models typically require large-scale, high-quality training data [28], motivating more scalable and diverse simulation.

C. Generative Simulation.

Generative simulation scales embodied AI by automatically creating environments, assets, and tasks. Representative systems include Gen2Sim [8], which generates assets, task specifications, and rewards; RoboGen [9], which iteratively creates robotic tasks and environments; Genesis [7], a unified platform for generating multi-modal simulation data from language prompts; and ReGen [29], which studies inverse simulation design. Although these systems have shown promise in manipulation and control, generative simulation remains underexplored for visual navigation, where rendering fidelity and environmental diversity are critical. Our work addresses this gap with a generative simulation framework and a dynamic multi-object navigation benchmark.

III. Simulation Method

In this section, we present **Gen-NAV**, a generative simulation framework for visual navigation built upon 3DGS. As illustrated in Figure 2, Gen-NAV consists of two core components. The **Gaussian-based Simulator** (Section III-A)

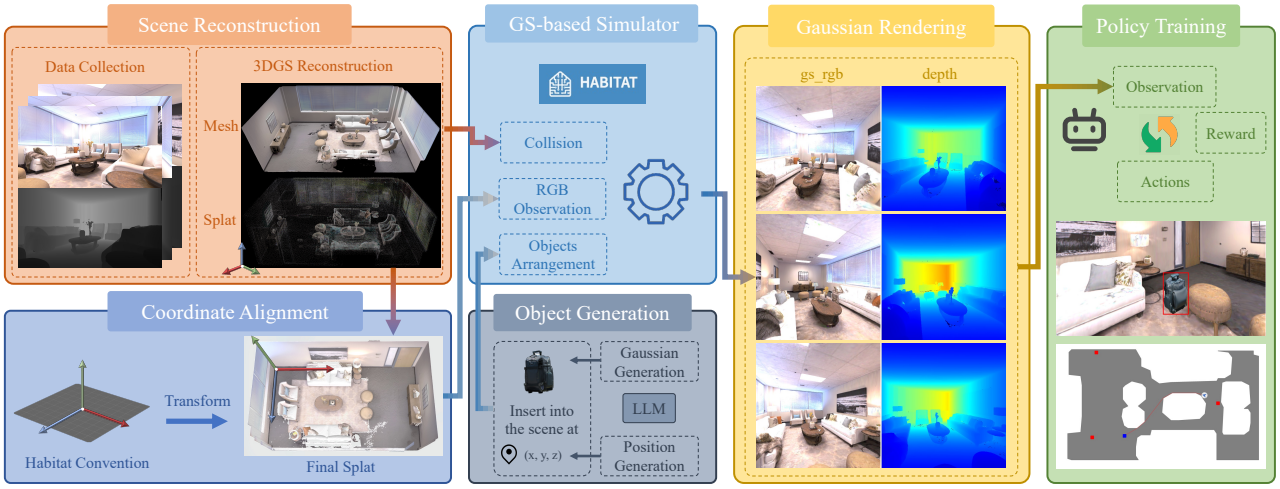


Fig. 2. **Overview of Gen-NAV.** Gen-NAV reconstructs indoor environments from RGB-D inputs using 3DGS and integrates them into Habitat via a coordinate alignment module. An object generation module subsequently synthesizes and places diverse object instances into the scene, enabling photorealistic and dynamic navigation environments.



Fig. 3. **Comparison of existing scenes and ours.** Scenes in HM3D exhibit common mesh artifacts such as holes and surface fragmentation, while our 3DGS-reconstructed scenes preserve finer visual details and produce smoother renderings.

reconstructs indoor scenes using 3DGS and supports real-time photorealistic RGB rendering with accurate geometry for navigation. The **Object Generative Module** (Section III-B) synthesizes diverse object instances and determines their context-aware placement and scale within the environment.

A. Gaussian-based Simulator

Indoor Scene Reconstruction. Existing indoor datasets face a fundamental trade-off between scalability and visual quality. Large-scale datasets such as MP3D [4] and HM3D [5] rely on scanning-based mesh reconstructions to achieve broad coverage, but often suffer from degraded visual fidelity. In contrast, the Replica dataset [3] provides high-quality reconstructions at the cost of significant manual effort, limiting its scalability. For generative simulation in VLN, both visual realism and scalable scene reconstruction are critical, as agents must perceive fine-grained visual cues while being trained across diverse environments. To address this challenge, we adopt 3DGS for indoor scene reconstruction, enabling fast image-based reconstruction with superior rendering quality. As shown in Figure 3, scenes reconstructed with 3DGS outperform those from HM3D, which frequently exhibit artifacts such as holes and geometric fragmentation.

Given a sequence of synchronized RGB images and depth maps $\{I_i, D_i\}$ captured from the target scene, our framework applies 3DGS reconstruction to obtain a Gaussian scene representation G_{room}. In parallel, we extract a mesh representation M_{room} using the VDBFusion algorithm [30] to support geometric reasoning.

This dual representation decouples visual fidelity from geometric accuracy: Gaussian representations are used for high-quality rendering, while meshes provide reliable geometry for navigation and collision handling. The framework is modular and supports seamless integration of different 3DGS reconstruction and mesh extraction methods. While vanilla 3DGS relies solely on RGB supervision and may suffer from floaters and incomplete surfaces, accurate geometry is essential for navigation. Following [31], we incorporate depth and normal constraints during optimization to improve geometric fidelity and surface consistency.

Coordinate Alignment. The reconstructed scenes are integrated into VLN simulators through coordinate alignment. We use Habitat [32] as an example to demonstrate the integration.

Simulation with Gaussian Rendering. After coordinate alignment, we extend the Habitat simulator with Gaussian-based rendering to enable photorealistic observations. The simulator consists of two core components: collision detection and visual observation acquisition. Collision detection is performed using the reconstructed mesh M_{room}, which provides sufficient geometric accuracy despite its limited rendering quality. For visual observations, we render RGB images directly from the aligned Gaussian scene representation G_{room}, achieving photorealistic and temporally consistent views for VLN agents.

B. Object Generation

As illustrated in Figure 4, the Object Generation Module comprises two primary components: object generation and object placement. The object generation component enables

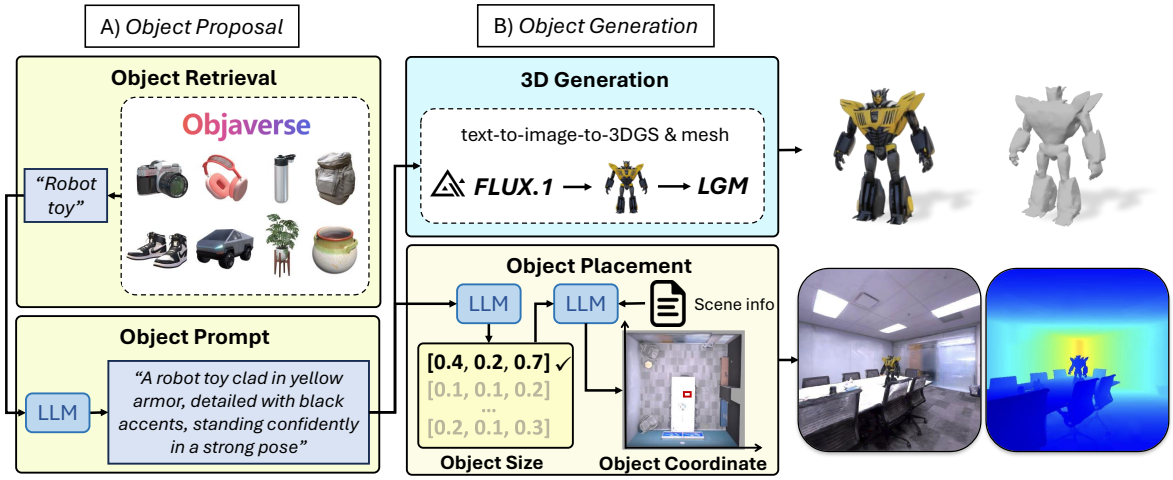


Fig. 4. **Overview of the Object Generation Module.** Object categories are selected from Objaverse, and category-specific descriptions are generated using an LLM. These descriptions are used to synthesize object images via text-to-image generation, which are then converted into 3D Gaussian Splatting representations and meshes using LGM. Since the generated objects are normalized in scale, an LLM is further employed to infer realistic object dimensions and determine context-aware placement locations within the scene.

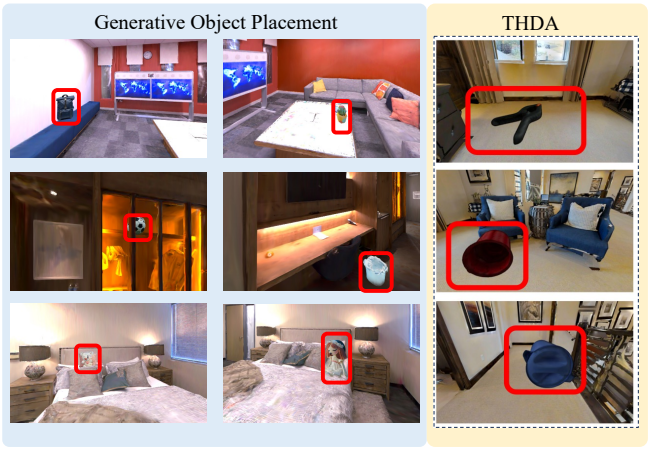


Fig. 5. **Comparison of object placement.** THDA inserts objects into scenes without explicit reasoning about object scale and spatial context, often resulting in implausible placements. In contrast, our method accounts for object dimensions and scene geometry, producing placements that better align with the surrounding environment.

the synthesis of arbitrary semantic object categories, while the placement component handles context-aware positioning and scaling of objects within the environment to ensure visual plausibility and functional realism.

Gaussian Generation of Objects. The first challenge in object generation is determining the types of objects needed. To this end, we use an LLM to select the top-k object categories suitable for indoor environments from the Objaverse taxonomy. Rather than directly using Objaverse assets, we adopt a generative pipeline to synthesize novel object instances, enabling open-vocabulary object diversity beyond predefined asset libraries. We then use GPT-4 [33] to generate detailed category-specific object descriptions. For object generation, we design a text-to-image-to-3DGS & mesh generation pipeline. We initially generate images of objects using FLUX.1-dev [34] based on their descriptions. Subsequently, the generated images are fed into the LGM [35] to produce the 3DGS G_{obj} of the objects, ensuring alignment with the 3DGS of indoor scenes. These 3DGS are then converted into meshes M_{obj} for subsequent collision testing.

Size and Position Generation of Objects. Appropriate object scale and placement are essential for realistic scene integration. We query GPT-4 with detailed object descriptions to obtain plausible size estimates, from which we probabilistically sample a scale configuration. For spatial placement, we construct a Bird's Eye View (BEV) representation containing scene coordinates and bounding boxes of existing objects. This information, together with the target object's semantic category and scale, is provided to GPT-4 to generate 2D BEV coordinates (x, y). Given the exported scene mesh, we compute feasible height values at the selected location and stochastically sample the final insertion height, while applying an upper-bound threshold to avoid ceiling-level placements. We also explored direct 3D coordinate generation (x, y, z) with GPT-4, but it often produced physically implausible positions. As shown in Figure 5, our method produces more reasonable object placements than THDA [36] in both position and scale.

IV. Navigation Task in Evolving Environments

A. Task Formulation

We define **Lifelong Navigation in Evolving Environments** as a sequential object navigation task, where an agent must navigate to a series of target objects within a single episode. Each target is specified by a semantic category descriptor, and the episode is decomposed into a sequence of subtasks $\{t_k\}_{k=1}^{N}$, with each subtask requiring the agent to locate and reach a designated target object g_k.

At the beginning of an episode, a set of target objects $\{g_i\}_{i=1}^{N}$ and distractor objects $\{d_j\}_{j=1}^{M}$ are sampled and placed in the environment subject to predefined constraints

on position and scale. The agent is initialized at a random location and starts with the first subtask t_1, which involves navigating to the proximity of target object g_1. For each subsequent subtask t_k, the agent navigates from its current position to the next target object g_k. Crucially, after the completion of each subtask, the environment is modified by rearranging the spatial configurations of the distractor objects. Object rearrangement occurs *between* subtasks rather than during continuous navigation, resulting in an evolving environment across subtasks while preserving stable low-level dynamics within each navigation stage. This design requires agents to repeatedly re-localize targets and adapt their spatial representations over a long horizon. The agent operates in a discrete action space $\mathcal{A} = \{\text{MoveForward}, \text{TurnLeft}, \text{TurnRight}, \text{LookUp}, \text{LookDown}, \text{Stop}\}$. Following GOAT-Bench [37], we augment this space with a `SubTaskStop` action to explicitly indicate subtask completion. Once `SubTaskStop` is executed, the current subtask t_k is terminated, the environment is updated, and the agent proceeds to the next subtask t_{k+1} from its current location.

Figure 1 illustrates a representative episode. Target objects (highlighted in green) serve as sequential navigation goals, while distractor objects (highlighted in red) are rearranged between subtasks. An agent is required to complete N consecutive subtasks by localizing a sequence of semantically distinct objects (e.g., headphones, camera, and LEGO toy) under evolving environment configurations.

B. Comparison with Other Tasks

The proposed task formulation differs from existing navigation benchmarks along several key dimensions. GOAT-Bench [37] introduced multi-goal navigation within a single episode, but operates on a fixed set of objects present in the original scene data. PIN [20] incorporates personalized instances as navigation targets and distractors, but remains limited to a single-goal setting. Similarly, HM3D-OVON [19] expands object category diversity in Object Navigation, while maintaining static environments and single-goal episodes.

In contrast, our task combines sequential multi-goal navigation with environment evolution across subtasks. By leveraging the generative capabilities of Gen-NAV, it supports navigation over diverse and uncommon object categories while introducing dynamic object rearrangements between goals. This unified formulation evaluates an agent's ability to perform long-horizon navigation under changing environment configurations, placing additional demands on spatial reasoning, memory, and continual adaptation.

V. Experiments

In this section, we first describe the construction of the Gen-NAV benchmark. We then evaluate representative modular and end-to-end navigation baselines on this benchmark and analyze their performance. Finally, we present ablation studies to examine the impact of key components in the proposed framework.

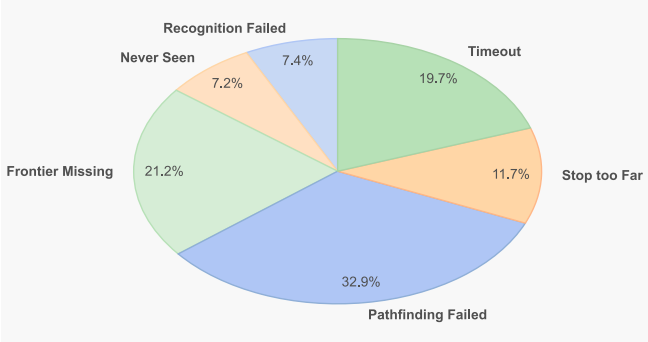


Fig. 6. **Aggregate failure causes on val-seen.** Failure modes are aggregated over complete episodes.

A. Gen-NAV Benchmark and Baseline

Benchmark. Based on the proposed Gaussian-based simulator and multi-object navigation task, we construct the Gen-NAV benchmark. We reconstruct high-fidelity indoor scenes from Replica [3] RGB-D data using DN-Splatter [31], producing aligned Gaussian and mesh representations. Using the Object Generation Module, we synthesize a diverse object set consisting of 28 uncommon semantic categories with multiple instances per category, resulting in 200 unique object instances. Each episode contains three target objects and three distractors, forming three sequential subtasks. We split Gen-NAV into training and four validation sets based on whether scenes and target goals are seen during training, as shown in Table I. These splits evaluate generalization to unseen objects, unseen scenes, and their combination.

Baselines and Metrics. We evaluate a modular approach, VLFM [38], and an end-to-end approach, OVON-DAgger [19]. Both methods are originally designed for single-object navigation in static environments. We report subtask-level Success Rate (SR) and Success weighted by Path Length (SPL) [16]. Results are summarized in Table II.

B. Baseline Performance and Analysis

Both modular and end-to-end baselines obtain low SR and SPL on the Gen-NAV benchmark, as shown in Table II. The val-unseen split is particularly challenging for both methods. For VLFM, SR decreases from 4.86 on val-seen to 1.83 on val-unseen, while for OVON-DAgger, SR decreases from 8.76 to 1.99. This indicates that generalization becomes more difficult when both scenes and target goals are unseen during training. Since the evaluated baselines were originally designed for static, single-object navigation, the current results should be interpreted as evaluating their robustness under the combined Gen-NAV setting, rather than isolating the effect of object rearrangement alone.

Figure 6 summarizes the aggregate distribution of final failure modes on the val-seen split. We use this analysis as a diagnostic example of common failure patterns, rather than as a split-wise comparison across all validation settings. On this split, a significant portion of failures arise from recognition-related issues, where target objects are either not detected or never observed. This suggests that object-centric perception

TABLE I

BENCHMARK SPLIT CHARACTERISTICS.

Split	Scenes seen during training	Goals seen during training	Episodes per scene
train	✓	✓	500
val-seen	✓	✓	50
val-unseen-objects	✓	✗	50
val-unseen-scenes	✗	✓	50
val-unseen	✗	✗	50

TABLE II

EVALUATION RESULTS. COMPARISON OF END-TO-END AND MODULAR METHODS ON 4 EVALUATION SPLITS.

	val-seen		val-unseen-objects		val-unseen-scenes		val-unseen	
Method	SR (↑)	SPL (↑)	SR (↑)	SPL (↑)	SR (↑)	SPL (↑)	SR (↑)	SPL (↑)
VLFM	4.86	3.62	4.86	3.42	3.50	2.32	1.83	1.24
OVON-DAgger	8.76	2.96	8.66	3.24	2.33	1.00	1.99	0.79

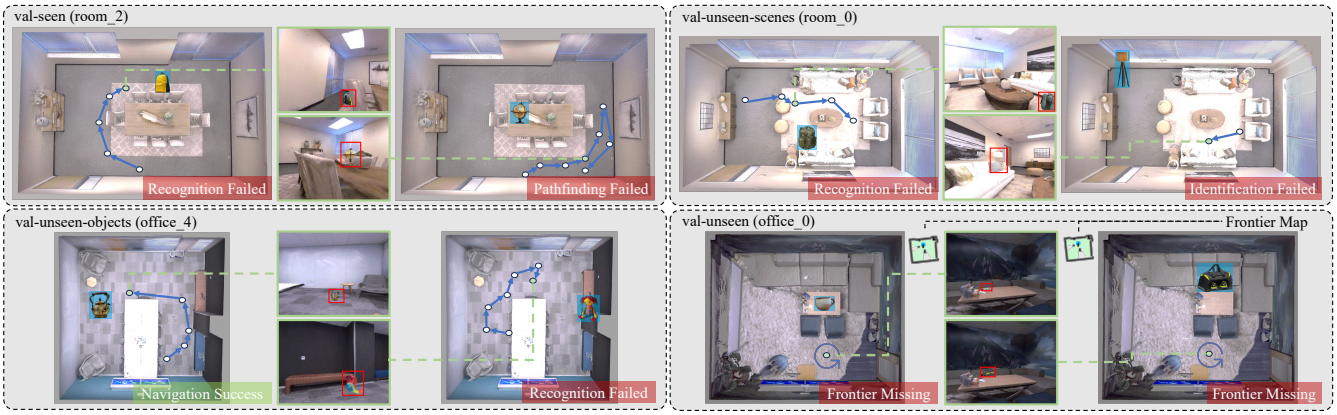


Fig. 7. **Qualitative navigation results across evaluation splits.** The agent is tasked with navigating to target objects such as backpacks, a globe, a lamp, a kettle, a rag doll, a cup, and a bag.

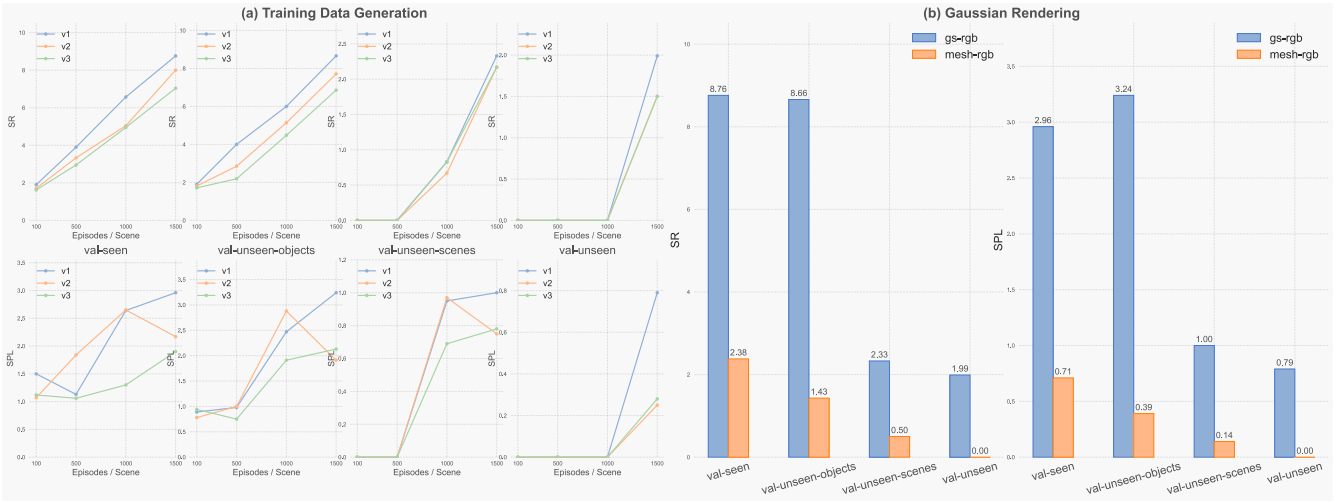


Fig. 8. **Ablation results.** We evaluate the effects of **(a) training data generation** and **(b) Gaussian rendering**.

models trained on static environments may struggle when objects are inserted, reconfigured, or appear in uncommon forms.

Another prevalent failure pattern involves premature termination due to missing frontiers or ineffective exploration. These cases highlight limitations in exploration strategies that implicitly assume static scene layouts, leading to suboptimal coverage once object configurations change across

TABLE III

ABLATION RESULTS FOR OBJECT SIZE ADJUSTMENT.

Method	Avg BLIP2 Score
w/ Size Adjustment	**0.4727**
w/o Size Adjustment	0.4567

subtasks. In addition, pathfinding failures and stop errors indicate difficulties in maintaining consistent spatial representations over long horizons, particularly when previously visited areas become outdated due to environment evolution. Since the current failure analysis aggregates final failure causes over complete episodes, a finer-grained subtask-wise analysis would further clarify how environment evolution and accumulated errors affect later stages of navigation.

C. Qualitative Results

Figure 7 presents representative navigation examples across different evaluation splits. In val-seen and val-unseen-objects settings, agents frequently observe target objects but fail to recognize them due to occlusion, small object size, or peripheral placement. In val-unseen-scenes, failures are often caused by ineffective exploration or incorrect path planning, resulting in agents terminating after exhausting frontier regions.

Notably, even when target objects are successfully detected, agents may still fail to complete navigation due to accumulated localization or planning errors across subtasks. These qualitative observations further illustrate the compounding difficulty introduced by sequential goals and evolving scene configurations.

D. Ablation Studies

Object Size Adjustment. We evaluate the effect of object size adjustment by comparing scale-aware insertion with direct insertion of generated objects at their normalized scales. We compute the BLIP2 Score as an image-text alignment score between the rendered image and the corresponding target-object text prompt. As shown in Table III, resizing improves the average BLIP2 Score, indicating better visual coherence. Although the improvement is moderate due to the limited spatial footprint of inserted objects, the results confirm the importance of scale-aware object integration.

Training Data Generation. To examine the effect of training data scale, we vary the number of training episodes per scene (100, 500, 1000, and 1500) and the number of distractors. OVON-DAgger is trained under identical settings and evaluated on the same test splits. Figure 8(a) shows that increasing training data generally improves performance, with the most notable gains observed on the val-unseen split. This suggests that generative simulation can partially mitigate generalization challenges by exposing agents to a broader distribution of object configurations during training.

Gaussian Rendering. We further compare training with RGB observations rendered using Gaussian Splatting and mesh-based rendering. As shown in Figure 8(b), Gaussian-rendered observations consistently yield better performance. This result indicates that higher visual fidelity facilitates object recognition and spatial reasoning, which are critical for navigation in evolving environments.

VI. CONCLUSION AND FUTURE WORK

In this work, we introduced **Gen-NAV**, a generative simulation framework that enables photorealistic and dynamic environments for visual navigation. By leveraging 3D Gaussian Splatting for high-fidelity scene reconstruction and generative object synthesis for open-vocabulary environment augmentation, Gen-NAV provides a scalable alternative to mesh-based simulators with limited realism and flexibility.

Built upon this framework, we proposed **Lifelong Navigation in Evolving Environments**, a navigation paradigm that requires agents to sequentially reach multiple object goals under evolving scene configurations. Our experiments show that existing modular and end-to-end navigation methods obtain low performance in this combined setting, highlighting challenges in object recognition, exploration, and long-horizon spatial reasoning. While the current evaluation does not isolate the individual contribution of each factor, it establishes a benchmark for studying navigation under more realistic and changing environments.

Future work includes controlled analyses of object rearrangement, subtask-wise failure attribution, more diverse scene generation, and extending Gen-NAV toward 4D scene simulation with dynamic geometry, controllable appearance, articulated objects, and temporally coherent interactions [39]–[44]. We hope Gen-NAV serves as a useful platform for advancing adaptive embodied navigation beyond static benchmarks.

REFERENCES

[1] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi, "Procthor: Large-scale embodied ai using procedural generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5982–5994, 2022.

[2] M. Khanna, Y. Mao, H. Jiang, S. Haresh, B. Shacklett, D. Batra, A. Clegg, E. Undersander, A. X. Chang, and M. Savva, "Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 384–16 393.

[3] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe, "The Replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[4] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *International Conference on 3D Vision (3DV)*, 2017.

[5] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra, "Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. [Online]. Available: https://openreview.net/forum?id=-v4OuqNs5P

[6] B. Shen, F. Xia, C. Li, R. Martín-Martín, L. Fan, G. Wang, C. Pérez-D'Arpino, S. Buch, S. Srivastava, L. Tchapmi, M. Tchapmi, K. Vainio, J. Wong, L. Fei-Fei, and S. Savarese, "igibson 1.0: A simulation environment for interactive tasks in large realistic scenes," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 7520–7527.

[7] G. Authors, "Genesis: A universal and generative physics engine for robotics and beyond," December 2024. [Online]. Available: https://github.com/Genesis-Embodied-AI/Genesis

[8] P. Katara, Z. Xian, and K. Fragkiadaki, "Gen2sim: Scaling up robot learning in simulation with generative models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6672–6679.

[9] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan, "Robogen: Towards unleashing infinite data for automated robot learning via generative simulation," *arXiv preprint arXiv:2311.01455*, 2023.

[10] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.

[11] N. Wang, L. Xiao, Y. Chen, W. Xiao, P. Merriaux, L. Lei, Z. Yan, S. Zhang, S. Xu, B. Li *et al.*, "Unifying appearance codes and bilateral grids for driving scene gaussian splatting," *Advances in Neural Information Processing Systems*, vol. 38, pp. 29 827–29 858, 2026.

[12] S. Zhang, B. Ye, X. Chen, Y. Chen, Z. Zhang, C. Peng, Y. Shi, and H. Zhao, "Drone-assisted road gaussian splatting with cross-view uncertainty," *arXiv preprint arXiv:2408.15242*, 2024.

[13] B. Ye, M. Qin, S. Zhang, M. Gong, S. Zhu, H. Zhao, and H. Zhao, "Gs-occ3d: Scaling vision-only occupancy reconstruction with gaussian splatting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 25 925–25 937.

[14] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, and S. Lee, "Sim-to-real transfer for vision-and-language navigation," in *Conference on Robot Learning*. PMLR, 2021, pp. 671–681.

[15] C. Li, F. Xia, R. Martín-Martín, M. Lingelbach, S. Srivastava, B. Shen, K. E. Vainio, C. Gokmen, G. Dharan, T. Jain, A. Kurenkov, K. Liu, H. Gweon, J. Wu, L. Fei-Fei, and S. Savarese, "igibson 2.0: Object-centric simulation for robot learning of everyday household tasks," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 455–465. [Online]. Available: https://proceedings.mlr.press/v164/li22b.html

[16] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, "On evaluation of embodied navigation agents," 2018. [Online]. Available: https://arxiv.org/abs/1807.06757

[17] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," 2020. [Online]. Available: https://arxiv.org/abs/2006.13171

[18] S. Wani, S. Patel, U. Jain, A. X. Chang, and M. Savva, "Multi-on: Benchmarking semantic map memory using multi-object navigation," in *Neural Information Processing Systems (NeurIPS)*, 2020.

[19] N. Yokoyama, R. Ramrakhya, A. Das, D. Batra, and S. Ha, "Hm3d-ovon: A dataset and benchmark for open-vocabulary object goal navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[20] L. Barsellotti, R. Bigazzi, M. Cornia, L. Baraldi, and R. Cucchiara, "Personalized instance-based navigation toward user-specific objects in realistic environments," in *Advances in Neural Information Processing Systems*, 2024.

[21] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4247–4258, 2020.

[22] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "Poni: Potential functions for objectgoal navigation with interaction-free learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 890–18 900.

[23] 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.

[24] S. Chen, T. Chabal, I. Laptev, and C. Schmid, "Object goal navigation with recursive implicit maps," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[25] R. Ramrakhya, D. Batra, E. Wijmans, and A. Das, "Pirlnav: Pretraining with imitation and rl finetuning for objectnav," in *CVPR*, 2023.

[26] A. Bar, G. Zhou, D. Tran, T. Darrell, and Y. LeCun, "Navigation world models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 15 791–15 801.

[27] H. Ren, Y. Zeng, Z. Bi, Z. Wan, J. Huang, and H. Cheng, "Prior does matter: Visual navigation via denoising diffusion bridge models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 12 100–12 110.

[28] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot, "Navigating to objects in the real world," *Science Robotics*, vol. 8, no. 79, p. eadf6991, 2023.

[29] P. Nguyen, T.-H. Wang, Z.-W. Hong, E. Aasi, A. Silva, G. Rosman, S. Karaman, and D. Rus, "Regen: Generative robot simulation via inverse design," *arXiv preprint arXiv:2511.04769*, 2025.

[30] I. Vizzo, T. Guadagnino, J. Behley, and C. Stachniss, "Vdbfusion: Flexible and efficient tsdf integration of range sensor data," *Sensors*, vol. 22, no. 3, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/1296

[31] M. Turkulainen, X. Ren, I. Melekhov, O. Seiskari, E. Rahtu, and J. Kannala, "Dn-splatter: Depth and normal priors for gaussian splatting and meshing," *arXiv preprint arXiv:2403.17822*, 2024.

[32] X. Puig, E. Undersander, A. Szot, M. D. Cote, R. Partsey, J. Yang, R. Desai, A. W. Clegg, M. Hlavac, T. Min, T. Gervet, V. Vondrus, V.-P. Berges, J. Turner, O. Maksymets, Z. Kira, M. Kalakrishnan, J. Malik, D. S. Chaplot, U. Jain, D. Batra, A. Rai, and R. Mottaghi, "Habitat 3.0: A co-habitat for humans, avatars and robots," 2023.

[33] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[34] B. F. Labs, "Flux," https://github.com/black-forest-labs/flux, 2023.

[35] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu, "Lgm: Large multi-view gaussian model for high-resolution 3d content creation," *arXiv preprint arXiv:2402.05054*, 2024.

[36] O. Maksymets, V. Cartillier, A. Gokaslan, E. Wijmans, W. Galuba, S. Lee, and D. Batra, "Thda: Treasure hunt data augmentation for semantic navigation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 354–15 363.

[37] M. Khanna*, R. Ramrakhya*, G. Chhablani, S. Yenamandra, T. Gervet, M. Chang, Z. Kira, D. S. Chaplot, D. Batra, and R. Mottaghi, "Goat-bench: A benchmark for multi-modal lifelong navigation," in *CVPR*, 2024.

[38] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *International Conference on Robotics and Automation (ICRA)*, 2024.

[39] T. Liu, Z. Chen, Z. Huang, S. Xu, S. Zhang, C. Ye, B. Li, Z. Cao, W. Li, H. Zhao *et al.*, "Light-x: Generative 4d video rendering with camera and illumination control," *arXiv preprint arXiv:2512.05115*, 2025.

[40] J. Weng, S. Zhang, Z. Diao, P. Li, H. Zhang, J. Chen, and H. Zhao, "Feedforward 3d editing learns from semantic-part transformation," *arXiv preprint arXiv:2605.27351*, 2026.

[41] J. Chen, B. Zhang, M. Chen, H. Zhang, S. Zhang, C. Zhu, H. Zhao, R. Huang, Z. Li, and Y. Wang, "One video, one world: Turning monocular video into physical 4d scenes," *arXiv preprint arXiv:2606.31388*, 2026.

[42] L. Shen, S. Zhang, H. Li, P. Yang, Z. Huang, Z. Zhang, and H. Zhao, "Gaussianart: Unified modeling of geometry and motion for articulated objects," in *2026 International Conference on 3D Vision (3DV)*. IEEE, 2026, pp. 384–395.

[43] Z. Huang, T. Liu, Z. Chen, S. Xu, S. Zhang, L. Xiao, Z. Cao, W. Li, H. Zhao, and Z. Liu, "Arthoi: Articulated human-object interaction synthesis by 4d reconstruction from video priors," *arXiv preprint arXiv:2603.04338*, 2026.

[44] H. Xu, S. Zhang, P. Li, B. Ye, X. Chen, H.-a. Gao, J. Zheng, X. Song, Z. Peng, R. Miao *et al.*, "Cruise: Cooperative reconstruction and editing in v2x scenarios using gaussian splatting," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 12 518–12 525.