ForageBench: A Photorealistic, Physically Grounded Benchmark for Interactive Object Search

Aparajito Saha[1] **Zhen Hao Gan**[1] **Jinjia Guo**[1] **Jacob Skwirsk**[1]

Jeremy Acheampong[1] **Anton Arapin**[1] **Chahyon Ku**[1] **Yue Hu**[1] **Nima Fazeli**[1]

Bernadette Bucher[1]

[1]University of Michigan, Ann Arbor

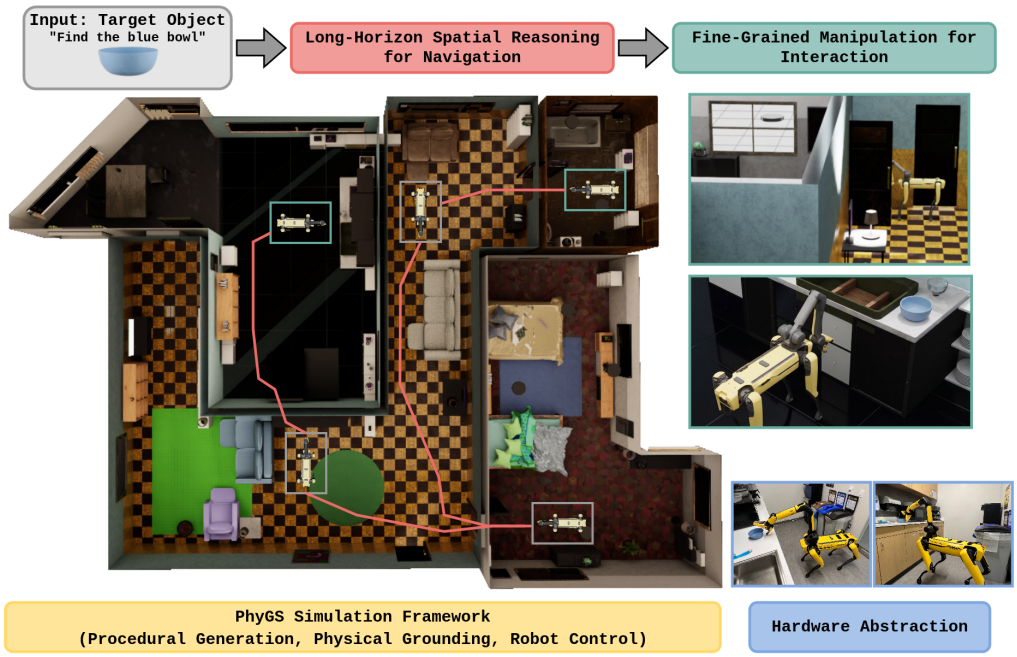

Figure 1: We develop **PhyGS**, a simulation framework in IsaacSim with a hardware abstraction of a Boston Dynamics Spot mobile manipulator. With PhyGS, we procedurally generate **ForageBench**, a benchmark for interactive object search on a mobile manipulator.

Abstract: Interactive semantic object search is the task of locating and retrieving a target object in an unseen, house-scale environment. This task requires both long-horizon spatial reasoning for navigation and fine-grained skills for manipulation, often demanding that an agent physically interact with the environment to make an occluded target perceivable. Prior benchmarks have largely isolated one capability or the other; we present ForageBench, a benchmark designed to study them jointly on a mobile manipulator developed in our newly introduced simulation framework PhyGS. PhyGS integrates large-scale procedural scene generation with photorealism, physical grounding, and low-level robotic control. Using PhyGS, we build an evaluation dataset of 25 house-scale scenes and 100 episodes spanning a progression of difficulty, and benchmark state-of-the-art vision-language models acting as high-level skill orchestrators. Our best baseline on a simulated Boston Dynamics Spot with Arm achieves an interactive search success of 4%. PhyGS provides

a hardware-abstracted interface mirroring Spot's SDK for development transfer between and simulation and the physical robot.

Keywords: Interactive Search; Mobile Manipulation; Simulation

1 Introduction

Consider the everyday request, "find my keys." A capable household robot must reason that the keys are likely on a counter or in a drawer, traverse a multi-room home it may never have seen, and physically open cabinets and move clutter to reveal an object that is not visible from any single vantage point. Perception here is not passive: the robot must *act in order to see*, deciding which receptacles to open and which objects to displace so that a target it has never directly observed becomes perceivable [1]. This is the problem of *interactive object search*: locating and retrieving a semantically specified target in an unseen, cluttered, house-scale environment, where success may depend on physically manipulating the environment to make the target perceivable [2, 3].

Despite its importance, interactive object search has resisted standardized study, because prior formulations have each relaxed one of the two properties that make the full task hard: house-scale spatial extent and physical interaction to find an unobserved target. The first family of work searches at *house scale but assumes the target is directly perceivable*. Object-goal navigation requires only that the agent reach an object that is visible from some vantage point, reducing the problem to collision-free motion and semantic frontier exploration [4, 5, 6, 7, 8]. Open-vocabulary and mobile-manipulator variants broaden the set of targets but retain the assumption that the target can ultimately be seen without altering the scene [9]. The second family permits *physical interaction but only at small scale*. Rearrangement and tabletop manipulation benchmarks assume the target location is already known and confine the agent to within arm's reach or a single room [10, 11, 12, 13]. The works that do search under occlusion treat it as manipulation-based active perception confined to a tabletop: the agent rummages through a bounded pile of reachable objects without moving its own base. Since the search space is small, it can be covered by near-exhaustive interaction rather than semantic targeting [14, 15, 16, 17]. Several recent methods extend a simplified version of interactive search to mobile manipulators in which scene information is gathered in advance, reducing interactive search to the problem of exploring a partially known environment [18, 3, 19]. Interaction is also limited to preset skills; for example, 1 of the 5 possible interaction skills in CuriousBot [19] is "lift clothes".

The regime we target is qualitatively harder than either of these. The agent must reason through interactive object search from onboard observations and a language goal, with no prior map. Interactions required to clear obstructions are *not* restricted to a preset list of manipulation skills given to the agent. Unlike tabletop rummaging, the candidate receptacles are house-scale and dispersed across rooms, so brute-force interaction is intractable and the agent must use semantic association to decide *where* to look before it can act. Unlike object-goal navigation, arriving at the right location is necessary but not sufficient, because the target may only be perceivable after environment interaction removes visual obstructions.

Two factors have historically kept this regime out of reach. First, choosing to interact with an unseen object on the basis of a language description and partial visual evidence requires open-vocabulary perception and commonsense semantic reasoning that have only recently become practical, through vision-language models and large language models used as perception front-ends and high-level reasoners [20, 21, 22, 23, 24, 25]. Second, studying the task in its full complexity demands an evaluation platform that is simultaneously photorealistic, physically interactable, controllably curated, and capable of low-level robotic control, a combination that no existing platform provides. Semantically rich embodied-AI simulators support reconfiguration of the scene but abstract away the continuous contact physics needed for realistic manipulation and sim-to-real transfer [26, 27, 28]; physics-driven engines provide high-fidelity contact dynamics but lack native support for generating house-scale photorealistic scenes [29, 30, 31, 32]; and recent agentic scene generators sacrifice the fine-grained controllability that standardized benchmarks demand [33, 34, 35, 36]. Building a

Figure 2: **PhyGS** is a simulation framework that enables the controllable generation of physically-grounded and visually diverse building-scale 3D scenes for simulating mobile manipulation tasks. The above image showcases several scenes generated by PhyGS and rendered in IsaacSim; we additionally provide a hardware abstraction layer for the Boston Dynamics Spot with Arm quadruped robot for validating low-level physical parameters and robot control in simulated scenes.

benchmark for interactive object search, therefore, first requires building a simulation framework that unifies all four capabilities in a single pipeline. The substantial engineering required has been one of the primary reasons why this task has not previously been studied in its full form.

Contributions. To this end, we present **ForageBench**, a photorealistic, physically grounded benchmark for interactive object search on mobile manipulators, together with the simulation framework, **PhyGS**, used to construct it. PhyGS extends the procedural scene generation of Infinigen-Indoors [37] within IsaacSim [29]: we make procedurally generated scenes physically interactive by intercepting articulated assets during export and re-injecting their joint parameters, augment the import process with rigid-body properties, collision meshes, and corrected lighting, and automate the population of scenes with manipulands from Objaverse-XL [38]. Using PhyGS, we construct ForageBench, an evaluation dataset of 25 house-scale scenes and 100 episodes. We develop a hardware abstraction layer for the Boston Dynamics Spot with Arm that mirrors the official SDK, enabling policies developed in simulation to transfer to the physical robot with minimal modification. We benchmark vision-language models (VLMs) [24, 25] acting as high-level orchestrators over a library of low-level skills on ForageBench. Our best baseline achieves a maximum interactive search success of only 4%. These results indicate that interactive object search is far from solved by modern robot foundation models or out-of-the-box skill orchestration, and they demonstrate ForageBench's value as a stress test for the next generation of embodied search agents.

2 Related Work

Semantic Object Search. Interactive search on a mobile manipulator requires large scale spatial reasoning and physical interaction to find an unobserved target. In search problems focusing on large scale spatial reasoning, the target is assumed to be directly perceivable from some viewpoint in the scene [4, 5] and leading methodological solutions are formulated as navigation problems [6, 7, 8]. Subsequent work extended these formulations to open-vocabulary targets and to mobile manipulator platforms while retaining the assumption that the target is observable without altering the scene [9]. In rearrangement and tabletop manipulation benchmarks, the target is visually occluded but the agent's search space is confined to within arm's reach or the same room [10, 11, 12, 13]. Due to the small search space, solutions target near-exhaustive interaction over semantic targeting [14, 15, 16, 17]. Several recent methods extend a simplified version of interactive search to mobile manipulators in which scene information is provided in advance, reducing interactive search to the problem of exploring a partially known environment with interaction limited to preset skills [18, 3, 19, 39]. In ForageBench, the agent must reason through interactive object search from onboard observations and a language goal, with no prior information about the scene and no preset skill list guaranteed to enable object discovery.

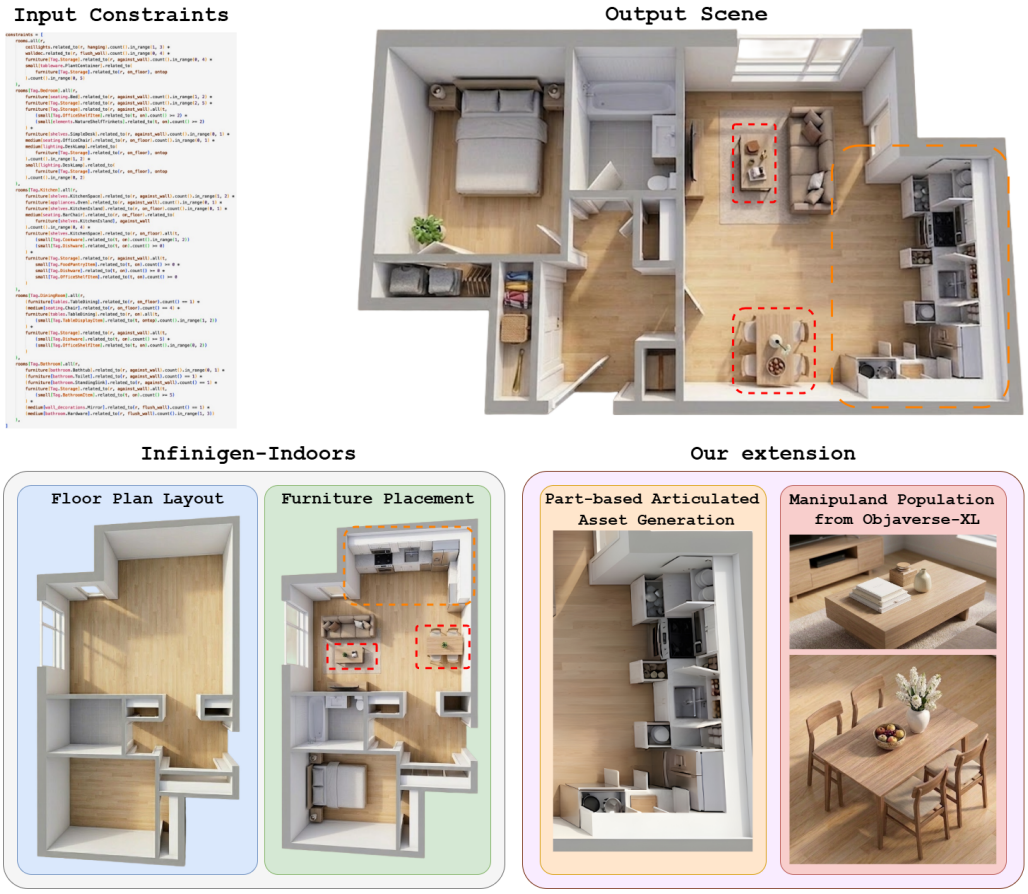

Figure 3: An end-to-end overview of the PhyGS generation process in Blender. Given a user-defined set of programmatic constraints for an indoor scene, PhyGS leverages Infinigen-Indoors for designing a valid floor plan and placing large furniture. We then extend the Infinigen engine to decompose assets that require articulation and add necessary joints, before populating the entire scene with labeled assets from Objaverse-XL based on semantic relevance and free volume.

Indoor Scene Generation. Simulation scene synthesis falls into three paradigms. Procedural methods [40, 37, 41] offer fine-grained control via hand-crafted rules but are restricted to closed-vocabulary asset libraries. Generative approaches [42, 43] learn spatial priors from human-generated 3D data [44, 45], producing realistic floor plans but struggling with multi-room layouts. LLM-driven agentic frameworks [33, 34, 35, 36, 46] capture complex distributions with open-vocabulary control, but their stochasticity precludes the fine-grained reproducibility required for standardized benchmarks. Thus, PhyGS extends the rule-based Infinigen-Indoors [37].

Embodied AI Simulation. Existing simulators trade semantic richness against physical fidelity. Semantically rich platforms [26, 10, 39] support high-level actions including container interaction but abstract away continuous control, while physics-driven platforms [29, 30, 31, 32] provide accurate contact dynamics but lack native house-scale photorealistic scene generation. BEHAVIOR-1K [28] is the closest hybrid but depends on a hand-curated asset library that bounds scale. External asset libraries are also manually curated [47, 48, 49], do not scale to large scenes [50], or lose articulation properties on export to downstream simulators [51]. PhyGS occupies a complementary point, pairing procedurally generated photorealistic scenes with IsaacSim contact dynamics in a single pipeline by intercepting articulated assets during procedural generation and re-injecting joint parameters at simulator export.

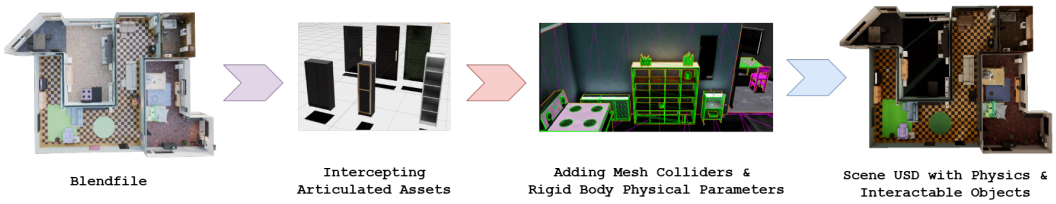

Figure 4: An overview of the scene export process for PhyGS. Infinigen-Indoors leverages Blender for procedural generation, but the default export process to 3D simulation environments sacrifices articulations, collision meshes and lighting. We design a custom exporter that intercepts articulated assets, adds joint parameters, populates all assets with mesh colliders and rigid body physical properties (e.g., mass and friction), and corrects the lighting scheme to retain textural and visual fidelity.

3 PhyGS: Physically-Grounded Controllable Scene Generation

Interactive search on a mobile manipulator requires multi-room spatial reasoning together with low-level control skills to interact with the environment. To train and evaluate algorithms for tasks with these simultaneous requirements, we require our simulation solution to have: 1) extensibility for diverse environment generation, 2) photorealistic rendering, 3) physical interactability, and 4) continuous robotic control. We introduce PhyGS to bridge photorealistic procedural scene generation in Infinigen [52] with robotic control in IsaacSim [29] and IsaacLab [30]. PhyGS automates the population of objects from Objaverse-XL [38] within these generated scenes.

Procedural Scene Generation We use *Infinigen-Indoors* [37] as our procedural engine due to its use of mathematical rules to create infinite variations of 3D scenes within the OpenUSD framework. We specifically employ *Infinigen* to construct underlying architectural elements and large furniture—components that are difficult to retrieve from standard datasets. Scenes are generated via configuration files for asset details and constraint files for spatial relations, ensuring that object arrangements respect physical plausibility (preventing mesh overlaps) and accessibility (ensuring free-space for manipulation).

Automated Articulation Interactive search tasks require agents to reason about physical obstructions, such as opening a cabinet to confirm the presence of a target object. While existing simulators like BEHAVIOR-1K [28] provide articulation, they often rely on hand-generated assets that limit scale and introduce procedural bias. In PhyGS, we intercept assets that require articulation and add necessary joints during scene generation in Blender.

A significant technical challenge arises during the export from Blender to USD: the standard baking process collapses articulated joints into static meshes. To overcome this, we developed a post-processing interceptor for the *Infinigen* exporter. Rather than manually redefining constraints for every scene, our system intercepts the parameters of articulated assets (doors, cabinets, and drawers) during the export phase as visualized in Figure 4. We preserve their joint geometry, material properties, and spatial relations, re-injecting this information into the final USD scene. This allows PhyGS to support robotic tasks requiring interaction with articulated objects without manual annotation or compromising the procedural pipeline.

Semantic Object Population Objects should be placed in semantically and spatially meaningful locations. We populate the procedurally generated base scenes with assets from Objaverse-XL [38] via an automated process visualized in Figure 3, selecting objects based on spatial and semantic relevance to indoor environment locations as well as photorealistic fidelity.

To ensure these objects are interactable for mobile manipulation, we assign them rigid-body physical properties and collision meshes within IsaacSim as visualized in Figure 4. While mass and friction can be randomized, we establish a baseline by fixing the static friction coefficient at 0.8 and setting the mass of interactable target objects to 1.0 kg, well within the 7.0 kg payload capacity of the Spot Arm. Unlike the standard footprint-based placement in *Infinigen*, our automated population

method accounts for volumetric availability, allowing for more complex arrangements and diverse grasp-pose selection scenarios.

Hardware Abstraction and Spot USD The final component of PhyGS is a high-fidelity model of the Boston Dynamics Spot equipped with a manipulation arm as a unified USD. To facilitate hardware-agnostic algorithm development and sim-to-real transfer, we developed an API that mirrors the official Boston Dynamics SDK. This abstraction of physical control enables policies developed in IsaacLab to be deployed on hardware with minimal modification as visualized in Figure 2.

4 ForageBench: An Interactive Search Benchmark

Building on the controllable scene generation capabilities of PhyGS, we introduce ForageBench, a standardized evaluation benchmark for the interactive object search task on mobile manipulators. ForageBench provides a curated dataset of house-scale episodes that span a range of physical and spatial difficulty, paired with a hardware-matched agent interface and a set of metrics designed to disentangle the contributions of navigation, manipulation, and semantic reasoning to overall task performance. By coupling rule-based procedural generation with rigid-body physics and articulated assets, ForageBench enables systematic, reproducible comparisons across embodied agent designs while preserving both visual and physical fidelity.

Task Definition. An agent is instantiated at a collision-free start pose within an unseen, house-scale indoor environment, with no prior knowledge of the scene's layout or the placement of objects within it. At the start of an episode, the agent is provided with a natural language description of a target object (e.g., *"a ceramic mug"*) that exists in the environment. It must locate and retrieve that object through a combination of navigation, exploration, and physical interaction with the environment. The agent perceives the environment through onboard RGB-D sensors and has access to the low-level controllers of the robot[1]. Success requires that the agent perceive the target object, physically retrieve it, and explicitly signal that it has done so. This may require manipulation of containing receptacles or occluding objects.

Dataset Generation. ForageBench consists of 25 single-story residential scenes generated procedurally with PhyGS, spanning two-, three-, and four-bedroom layouts that each additionally include a bathroom, kitchen, living room, and dining room. This range of layouts ensures that agents must reason over multi-room spatial structures while remaining within the computational budget of high-fidelity rendering for full-house environments. To support tasks that require interaction with the environment, we instantiate a diverse set of articulated furniture in each scene, including doors, cabinets, drawers, windows, ovens, and dishwashers in a closed configuration at each episode's start.

We populate each scene with gripper-scale manipulands drawn from Objaverse-XL [38], spanning categories such as kitchen utensils, statues, and bottles. Each manipuland is unique within a scene to avoid ambiguity in target specification. The total number of manipulands varies from 50 to 200 per scene depending on the floor plan size and a configurable density parameter.

Episodes. ForageBench comprises 100 episodes distributed across the 25 scenes, with each scene contributing four episodes that span a progression of task difficulty from easiest to hardest: **1. Target object in a separate room, but visible.** The agent must open doors to traverse between rooms, but does not need to open receptacles. **2. Target object in the same room, but not perceivable.** The agent must leverage semantic cues to localize the target and manipulate receptacles to retrieve it without leaving the room. **3. Target object in a separate room, not perceivable.** The agent must traverse between rooms *and* open receptacles over the course of its search. **4. Target object in a separate room, requires manipulation of non-target objects to perceive.** In addition to the requirements from 3, the agent must move non-target objects that occlude its view of the target before the target can be perceived and retrieved.

[1]The agent may invoke a fixed set of high-level skills exposed via an interface described in Agent Interface. However, calling these skills is an implementation decision, not an implementation requirement enforced by the benchmark which exposes access to continuous control of the robot.

At the start of each episode, the agent is instantiated at a configured collision-free start pose, and all articulated objects are set to their closed state. The simulator advances at 60 Hz of simulated time, though physics and rendering overhead reduce the effective wall-clock rate. An episode terminates in failure if any of the following conditions are met: collision, exiting the bounds of the house, selecting an incompatible skill, repeating the same skill three consecutive times, and issuing a `report_found` signal for an incorrect target object.

Agent Interface. To support reproducible evaluation and seamless sim-to-real transfer, ForageBench exposes a high-level skill API matched to the Boston Dynamics Spot with Arm SDK as an abstraction of the hardware platform. Skills may be invoked either synchronously or asynchronously, and each skill returns a structured response indicating success or a specific failure mode. The full set of skills available to the agent is: {`navigate(target)`, `place(receptacle_reference)`, `open_door(reference)`, `close_door(reference)`, `open_drawer(reference)`, `close_drawer(reference)`, `report_found(object)`}.

Perception is exposed through a unified API that mirrors the onboard sensor stack of the physical Spot. The agent retrieves RGB-D observations in the robot frame via `get_image_from_sources[source_name]`, and a set of transformation helpers convert detected object coordinates from the camera frame to the world frame for use in downstream planning.

Note that future methods which evaluate on this benchmark do not have to adhere to this agent interface and can develop either different component skills or autonomy stacks with different modularity designs. This agent interface is a hardware abstraction of the Boston Dynamics Spot robot, enabling hardware deployment on a popular mobile manipulation platform, not a requirement for methodological research on the benchmark.

Metrics. We report five metrics on ForageBench, each capturing a distinct facet of task performance:

- **Traversal success (TS):** the agent reaches the target object's location within a 0.6 m tolerance, with successful execution of any intervening manipulation skills required to access that location. The agent need not retrieve the target object for the traversal to be successful, but must issue either a `report_found` or `fail` signal from within this location.
- **Interactive object search success (IOSS):** all conditions for traversal success are met, and the agent additionally executes the manipulation actions necessary to retrieve the target object from within its containing receptacle, issuing `report_found` after a successful `pick`.
- **Skill execution success:** the percentage of skill invocations across episodes that execute without returning error codes. In this case, we do not consider whether the agent successfully completed the primary task; our focus is on evaluating underlying failures in skill execution arising from controller-level issues as opposed to high-level decision making.
- **Success weighted by path length (SPL):** the standard SPL metric from prior navigation benchmarks [4], where the path length includes any auxiliary motions taken to inspect or articulate scene elements. IOSS indicates the binary success for each episode, which we then weight by path length.
- **Success weighted by path length and manipulation efficiency (SPL-M):** an extension of SPL that incorporates manipulation actions into the cost of the optimal trajectory.

5 Experiments

A growing body of work uses large vision-language models (VLMs) as high-level orchestrators that select from a fixed library of low-level skills [20, 21, 23, 22, 53, 24, 25]. This architecture, a VLM orchestrator over a skill API, is precisely the baseline class we evaluate on ForageBench. We evaluate two high-level orchestrators [2] well-suited for the interactive object search task, Qwen3-VL-8B-Instruct [25] and Gemini Robotics ER-1.6 [24], under two conditions: a *known* target object

[2] Subskills called by the orchestrators are defined in the supplement

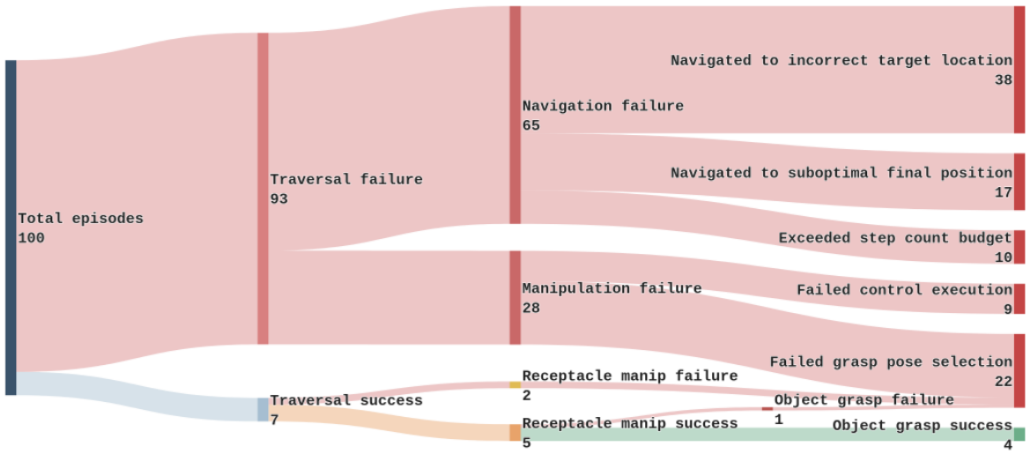

Figure 5: Success and failure cases in our experiments for the best-performing orchestrator (Gemini Robotics ER-1.6) over 100 episodes with the affordance prior included in the agent prompt and an unknown target object location. From left to right, we show the application of the components of our baseline agent and a breakdown of the long-tail failure modes of each of the components.

location, which provides ground-truth target coordinates alongside the target's semantics, and an *unknown* target location, which provides only the target object in the prompt.

Table 1: Summary of success metrics from using different large vision-language models as orchestrators for ForageBench. Each orchestrator is run on the full set of 100 episodes per condition.

	Known target object location				**Unknown target object location**			
Orchestrator	TS (%)	IOSS (%)	SPL (%)	SPL-M (%)	TS (%)	IOSS (%)	SPL (%)	SPL-M (%)
Qwen3	13.00	8.00	5.22	3.47	2.00	2.00	0.39	0.11
Gemini Robotics	15.00	11.00	6.08	4.64	7.00	4.00	0.55	0.35

Results. Table 1 reports performance under both conditions using four metrics defined previously. Gemini Robotics outperforms Qwen3 on every metric in both settings. The strongest agent completes only 11% of episodes (IOSS) when given a ground truth target location which drops to 4% when the location is unknown. The gap between TS and IOSS indicates that reaching the target is necessary but not sufficient; a non-trivial share of episodes that navigate successfully still fail during manipulation. The path-length-weighted metrics are far below the corresponding raw success rates, showing that successful episodes are typically reached via inefficient trajectories; this is most pronounced in the unknown-location condition, where SPL falls to 0.55% for Gemini Robotics and 0.39% for Qwen3.

Failure Analysis. Figure 5 decomposes the 100 episodes of the Gemini Robotics agent under the unknown-location condition into a hierarchy of failure modes. Traversal is the dominant bottleneck: 93 episodes fail before completing the final manipulation stage, mainly due to the agent being unable to navigate to a location that corresponds to the target object's semantics. Failures due to suboptimal final positioning result in an inability to perceive and manipulate target receptacles. Exceeding the episode's step budget arises from the agent roaming through the household. Our baseline agent suffers from grasp pose selection failures from AO-Grasp, which is limited to an overall success rate of 37.4% and is optimized for parallel jaw grippers [54], as opposed to Spot's hinged gripper which requires more nuanced grasp poses. Only 7 episodes reach within the target location for the object, of which 5 successfully manipulate the receptacle and 4 complete the final grasp.

6 Limitations

PhyGS. Our simulation capability has many areas for potential improvement which we will continue to extend. In PhyGS, our use of ray-traced lighting within IsaacSim achieves high visual fidelity. However, the computational overhead of real-time rendering remains a constraint for massive parallelization of RL policies on consumer-grade hardware. Collision checking from our physics simulator substantially slows down rendering of larger scale scenes. We plan to continue improving this rendering speed in future releases of our simulation capability.

ForageBench. Our analysis focuses solely on leveraging VLMs for skill orchestration. Works that construct language-grounded dynamic scene graphs [3] and open-set task-driven scene graphs [55, 56] could theoretically support efficient hierarchical search for improving reasoning for this task. We view adaptions of these methods as natural candidates for future evaluation on ForageBench.

References

[1] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988. doi:10.1109/5.5968.

[2] K. Zheng. *Generalized Object Search*. PhD thesis, Brown University, February 2023.

[3] D. Honerkamp, M. Büchner, F. Despinoy, T. Welschehold, and A. Valada. Language-grounded dynamic scene graphs for interactive object search with mobile manipulation. *IEEE Robotics and Automation Letters*, 2024.

[4] 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. *arXiv preprint arXiv:2006.13171*, 2020.

[5] K. Yadav, S. K. Ramakrishnan, J. Turner, A. Gokaslan, O. Maksymets, R. Jain, R. Ramrakhya, A. X. Chang, A. Clegg, M. Savva, E. Undersander, D. S. Chaplot, and D. Batra. Habitat challenge 2022. `https://aihabitat.org/challenge/2022/`, 2022.

[6] 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. *CVPR*, 2023.

[7] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang. ESC: Exploration with Soft Commonsense Constraints for Zero-shot Object Navigation. *arXiv preprint arXiv:2301.13166*, 2023.

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

[9] S. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. Turner, Z. Kira, M. Savva, A. Chang, D. S. Chaplot, D. Batra, R. Mottaghi, Y. Bisk, and C. Paxton. Homerobot: Open-vocabulary mobile manipulation. arXiv:2306.11565, 2024.

[10] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[11] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi. Visual room rearrangement. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[12] C. Gan, S. Zhou, J. Schwartz, S. Alter, A. Bhandwaldar, D. Gutfreund, D. L. Yamins, J. J. DiCarlo, J. McDermott, A. Torralba, and J. B. Tenenbaum. The threedworld transport challenge: A visually guided task-and-motion planning benchmark towards physically realistic embodied ai. IEEE International Conference on Robotics and Automation, 2022.

[13] K. Ehsani, W. Han, A. Herrasti, E. VanderBilt, L. Weihs, E. Kolve, A. Kembhavi, and R. Mottaghi. ManipulaTHOR: A Framework for Visual Object Manipulation. In *CVPR*, 2021.

[14] M. R. Dogar, M. C. Koval, A. Tallavajhula, and S. S. Srinivasa. Object search by manipulation. In *2013 IEEE International Conference on Robotics and Automation*, pages 4973–4980, 2013. doi:10.1109/ICRA.2013.6631288.

[15] L. L. Wong, L. P. Kaelbling, and T. Lozano-Pérez. Manipulation-based active search for occluded objects. In *2013 IEEE International Conference on Robotics and Automation*, pages 2814–2819. IEEE, 2013.

[16] T. Novkovic, R. Pautrat, F. Furrer, M. Breyer, R. Siegwart, and J. Nieto. Object finding in cluttered scenes using interactive perception. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8338–8344. IEEE, 2020.

[17] H. Jiang, B. Huang, R. Wu, Z. Li, S. Garg, H. Nayyeri, S. Wang, and Y. Li. Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation. *arXiv preprint arXiv:2402.15487*, 2024.

[18] F. Schmalstieg, D. Honerkamp, T. Welschehold, and A. Valada. Learning hierarchical interactive multi-object search for mobile manipulation. *IEEE Robotics and Automation Letters*, 2023.

[19] Y. Wang, L. Fermoselle, T. Kelestemur, J. Wang, and Y. Li. CuriousBot: Interactive mobile exploration via actionable 3D relational object graph. *IEEE Robotics and Automation Letters*, 2026. doi:10.1109/LRA.2026.3666384. arXiv:2501.13338.

[20] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*, 2022.

[21] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence. PaLM-E: An embodied multimodal language model. In *International Conference on Machine Learning (ICML)*, 2023.

[22] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

[23] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei. VoxPoser: Composable 3d value maps for robotic manipulation with language models. In *Conference on Robot Learning (CoRL)*, 2023.

[24] Google DeepMind. Gemini Robotics ER-1.6. URL https://deepmind.google/models/gemini-robotics/gemini-robotics-er/.

[25] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, W. Ge, Z. Guo, Q. Huang, J. Huang, F. Huang, B. Hui, S. Jiang, Z. Li, M. Li, M. Li, K. Li, Z. Lin, J. Lin, X. Liu, J. Liu, C. Liu, Y. Liu, D. Liu, S. Liu, D. Lu, R. Luo, C. Lv, R. Men, L. Meng, X. Ren, X. Ren, S. Song, Y. Sun, J. Tang, J. Tu, J. Wan, P. Wang, P. Wang, Q. Wang, Y. Wang, T. Xie, Y. Xu, H. Xu, J. Xu, Z. Yang, M. Yang, J. Yang, A. Yang, B. Yu, F. Zhang, H. Zhang, X. Zhang, B. Zheng, H. Zhong, J. Zhou, F. Zhou, J. Zhou, Y. Zhu, and K. Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

[26] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, A. Kembhavi, A. Gupta, and A. Farhadi. Ai2-thor: An interactive 3d environment for visual ai. arxiv:1712.05474, 2017.

[27] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[28] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning (CoRL)*, 2023.

[29] NVIDIA. Isaac Sim. URL https://github.com/isaac-sim/IsaacSim.

[30] M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Munoz, X. Yao, R. Zurbrügg, N. Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. In *arXiv preprint arXiv:2511.04831*, 2025.

[31] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi:10.1109/IROS.2012.6386109.

[32] R. Tedrake and the Drake Development Team. Drake: Model-based design and verification for robotics, 2019. URL https://drake.mit.edu.

[33] Y. Yang, F.-Y. Sun, L. Weihs, E. VanderBilt, A. Herrasti, W. Han, J. Wu, N. Haber, R. Krishna, L. Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[34] Y. Wang, X. Qiu, J. Liu, Z. Chen, J. Cai, Y. Wang, T.-H. Wang, Z. Xian, and C. Gan. Architect: Generating vivid and interactive 3d scenes with hierarchical 2d inpainting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[35] Y. Yang, B. Jia, S. Zhang, and S. Huang. Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[36] H. Xia, X. Li, Z. Li, Q. Ma, J. Xu, M.-Y. Liu, Y. Cui, T.-Y. Lin, W.-C. Ma, S. Wang, S. Song, and F. Wei. Sage: Scalable agentic 3d scene generation for embodied ai. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[37] A. Raistrick, L. Mei, K. Kayan, D. Yan, Y. Zuo, B. Han, H. Wen, M. Parakh, S. Alexandropoulos, L. Lipson, et al. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[38] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, E. VanderBilt, A. Kembhavi, C. Vondrick, G. Gkioxari, K. Ehsani, L. Schmidt, and A. Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. In *arXiv preprint arXiv:2307.05663*, 2023.

[39] F. Xia, W. B. Shen, C. Li, P. Kasimbeg, M. E. Tchapmi, A. Toshev, R. Martín-Martín, and S. Savarese. Interactive Gibson benchmark: A benchmark for interactive navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 5(2):713–720, 2020.

[40] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[41] C. Eppner, A. Murali, C. Garrett, R. O'Flaherty, T. Hermans, W. Yang, and D. Fox. scene_synthesizer: A python library for procedural scene generation in robot manipulation. In *Journal of Open Source Software*, 2024.

[42] D. Paschalidou, A. Kar, M. Shugrina, K. Kreis, A. Geiger, and S. Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[43] Y. Yang, B. Jia, P. Zhi, and S. Huang. Physcene: Physically interactable 3d scene synthesis for embodied ai. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[44] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.

[45] 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. In *International Conference on 3D Vision (3DV)*, 2017.

[46] N. Pfaff, T. Cohn, S. Zakharov, R. Cory, and R. Tedrake. Scenesmith: Agentic generation of simulation-ready indoor scenes. In *arXiv preprint arXiv:2602.09153*, 2026.

[47] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su. Sapien: A simulated part-based interactive environment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[48] H. Geng, H. Xu, C. Zhao, C. Xu, L. Yi, S. Huang, and H. Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[49] Z. Jin, Z. Che, T. Li, Z. Zhao, K. Wu, Y. Zhang, et al. Artvip: Articulated digital assets of visual realism, modular interaction, and physical fidelity for robot learning. In *International Conference on Learning Representations (ICLR)*, 2026.

[50] H. Xia, E. Su, M. Memmel, A. Jain, R. Yu, N. Mbiziwo-Tiapo, A. Farhadi, A. Gupta, S. Wang, and W.-C. Ma. Drawer: Digital reconstruction and articulation with environment realism. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[51] A. Joshi, B. Han, J. Nugent, M. G. Saez-Diez, Y. Zuo, J. Liu, H. Wen, S. Alexandropoulos, K. Kayan, A. Calveri, T. Sun, G. Liu, Y. Shao, A. Raistrick, and J. Deng. Procedural generation of articulated simulation-ready assets. In *arXiv preprint arXiv:2505.10755*, 2025.

[52] A. Raistrick, L. Lipson, Z. Ma, L. Mei, M. Wang, Y. Zuo, K. Kayan, H. Wen, B. Han, Y. Wang, A. Newell, H. Law, A. Goyal, K. Yang, and J. Deng. Infinite photorealistic worlds using procedural generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[53] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, 2024. *URL https://arxiv. org/abs/2410.24164*.

[54] C. Parés-Morlans, C. Chen, Y. Weng, M. Yi, Y. Huang, N. Heppert, L. Zhou, L. J. Guibas, and J. Bohg. Ao-grasp: Articulated object grasp generation. *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13096–13103, 2023. URL https://api.semanticscholar.org/CorpusID:264439321.

[55] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone. Clio: Real-time task-driven open-set 3d scene graphs. *IEEE Robotics and Automation Letters*, 9(10):8921–8928, 2024. doi:10.1109/LRA.2024.3451395.

[56] Y. Chang, L. Fermoselle, D. H. Ta, B. Bucher, L. Carlone, and J. Wang. Ashita: Automatic scene-grounded hierarchical task analysis. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29458–29468, 2025. URL https://api.semanticscholar.org/CorpusID:277634120.