LIVIN: BENCHMARKING SPATIAL AND EMBODIED INTELLIGENCE IN DIGITAL TWINS OF LIVED-IN HOMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Realistic household simulation must capture not only diverse environments but also the lived-in object arrangements and spatial constraints that shape robot motion and interaction. Existing resources often trade off scale, real-world correspondence, and interaction readiness, leaving a gap in faithful, interactive replicas of how real homes are actually arranged. To this end, we introduce LIVIN, a benchmark for spatial and embodied intelligence built on digital twins of 30 diverse lived-in homes. These replicas preserve observed room layouts, furniture configurations, and everyday belongings. To construct them, we design a human-in-the-loop workflow comprising instance recognition, architectural reconstruction, and object generation and placement, with intermediate results reviewed and corrected by humans against the source observations at each stage. We evaluate four tasks in LIVIN: 3D detection, 3D reconstruction, navigation, and loco-manipulation. Our evaluations show that current methods remain challenged by the dense object arrangements, occlusions, limited free space, and constrained interaction regions found in realistic lived-in homes. We hope LIVIN will help advance embodied AI in real-world homes, from spatial understanding to robotic interaction, and ultimately bring embodied intelligence into everyday home environments.



Figure 1: **Overview of LIVIN.** Built from 30 lived-in homes, LIVIN captures realistic household layouts, furnishings, and daily-use objects, enabling evaluation of 3D detection, reconstruction, navigation, and loco-manipulation in realistic domestic environments.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Everyday homes impose dense and highly variable spatial constraints on robot behavior: furniture limits traversable space, belongings create clutter and occlusion, and surrounding objects affect how a target can be reached or manipulated. Household robots therefore require both *spatial intelligence* to understand objects, geometry, and spatial relationships, as well as *embodied intelligence* to act within these constraints. Environmental diversity is crucial for generalization; for example, $\pi_{0.5}$ reports improved performance as mobile-manipulation training scales from 3 to 104 locations (Intelligence et al., 2025). While simulation can scale such experience, constructing realistic, interactive replicas of diverse lived-in homes remains challenging.

Existing resources address different aspects of the problem. Artist-designed datasets and procedural generators provide diverse layouts and object configurations (Fu et al., 2021; Deitke et al., 2022; Raistrick et al., 2024), but are synthesized rather than grounded in observed homes. Other efforts such as MetaScenes and InternScenes (Yu et al., 2025; Zhong et al., 2026) convert real-world scans into simulatable scenes with detailed object arrangements. These pipelines, however, rely heavily on asset retrieval and provide limited coverage of articulated household objects. More recently, agentic systems such as SceneSmith (Pfaff et al., 2026) automate increasing portions of scene and asset construction. Together, these advances make it possible to move beyond approximate scene replicas toward observation-grounded digital twins that preserve *real household layouts, rich object instances, and interactable articulation*, enabling spatial understanding and embodied behavior to be evaluated in the challenging realistic environments.

We introduce **LIVIN**, a benchmark built on digital twins of **30 diverse lived-in homes** for jointly evaluating spatial and embodied intelligence. LIVIN contains 319 rooms and 12,808 object instances, including 1,559 articulated objects, reflecting the dense and varied object arrangements of real households (Sec. 3). Each home is reconstructed from floorplans, posed multi-view images, and metric scans, preserving room-scale geometry, furniture layouts, and everyday belongings, including small objects that create realistic occlusions and constrain navigation and interaction.

To construct these observation-grounded replicas, we develop a human-in-the-loop workflow comprising instance recognition, architectural reconstruction, and object generation and placement. We treat articulated and non-articulated objects with different reconstruction strategies. Articulated objects are jointly modeled and placed by GPT-6 Astra (OpenAI, 2026) through Blender code, preserving their kinematic structure while fitting the surrounding environment. Non-articulated objects are generated by Rodin (Zhang et al., 2024) from enhanced reference images and then placed to match the captured arrangements. Each construction stage is followed by human review, allowing intermediate results to be inspected and corrected against the source observations before proceeding.

LIVIN supports four benchmarks: 3D detection, 3D reconstruction, navigation, and loco-manipulation. The 3D benchmarks evaluate object localization and geometry recovery under clutter, occlusion, and dense household arrangements. The navigation benchmark targets long-horizon humanoid navigation, requiring robots to reason about routes and execute physically feasible whole-body motion through constrained spaces. The loco-manipulation benchmark further evaluates physical interaction, including pick-and-place, pushing, articulated-object manipulation, and long-horizon tasks that couple locomotion with dependent manipulation subgoals. Across these benchmarks, we find that existing methods still struggle in complex household scenes: 3D methods may miss objects or recover inaccurate geometry, while embodied systems may fail to localize the intended destination, escape obstacles, establish stable grasps, or complete long-horizon manipulation sequences.

The work is organized around three contributions:

- **LIVIN-Scenes: 30 diverse lived-in homes**, providing observation-grounded digital twins that preserve realistic household layouts, everyday object arrangements, and rich object categories under natural spatial constraints.
- **A human-in-the-loop reconstruction workflow grounded in captured observations**, combining agentic reasoning, code-based modeling, and image-conditioned 3D generation to reconstruct architectural structure, articulated objects, and everyday belongings with stage-wise human verification.

- **LIVIN-Bench: a unified evaluation suite for 3D and embodied tasks**, spanning 3D detection, 3D reconstruction, navigation, and loco-manipulation to evaluate scene understanding, robot mobility, and physical interaction in realistic household environments.

2 Related Work

Real-to-Sim Generation. Recent 3D generative models have substantially improved the conversion of real-world observations into reusable virtual assets and structured scenes. At the object level, methods such as TRELLIS, Clay, and SAR3D recover detailed geometry and appearance from sparse visual inputs (Xiang et al., 2025; 2026; Zhang et al., 2024; Chen et al., 2025), while some works further extend 3D generation to articulated and interaction-ready assets with explicit part structures and motion parameters (Liu et al., 2024; Zhou et al., 2026; Gao et al., 2025; Wang et al., 2026). At the scene level, recent methods such as MIDI and SceneGen generate compositional 3D environments by jointly modeling object geometry and scene layout from visual or structured conditions (Huang et al., 2025; Meng et al., 2026), while CAST, SAM3D and RecGen reconstruct 3D scenes by recovering object geometry and estimating object poses from visual observations (Yao et al., 2025; Chen et al., 2026b; Zadaianchuk et al., 2026). LIVIN extends this direction to complete lived-in homes through a human-in-the-loop agentic workflow that reconstructs architectural structure, scene layout, articulated objects, and high-fidelity everyday objects, producing more complete and faithful replicas of real household environments.

Indoor Scene Datasets. Existing indoor scene resources are built through several distinct approaches. Artist-designed datasets such as 3D-FRONT and HSSD offer high-quality structured interiors but require substantial manual effort (Fu et al., 2021; Khanna et al., 2024). Procedural methods such as ProcTHOR and Infinigen Indoors scale more easily (Deitke et al., 2022; Raistrick et al., 2024), but their rule-based layouts and reused assets limit correspondence to real homes. Recent agentic and generative systems further automate scene creation (Yang et al., 2024; Xia et al., 2026b; Pfaff et al., 2026; Qin et al., 2026; Chen et al., 2026a), but their outputs still require verification for scene-level consistency. Capture-grounded datasets such as Scan2CAD, ReplicaCAD, MetaScenes, and InternScenes derive virtual environments from real-world observations (Avetisyan et al., 2019; Szot et al., 2021; Yu et al., 2025; Zhong et al., 2026), yet retrieval, rearrangement, or partial reconstruction may not fully preserve instance-specific structure and complete object arrangements. In contrast, LIVIN is fully grounded in real lived-in homes, with reconstruction initialized by agents and subsequently aligned and corrected against real-world evidence, balancing efficiency and fidelity to produce detailed, articulated, interaction-ready replicas.

Navigation and Loco-manipulation Benchmarks. Existing embodied benchmarks span real scans, curated scenes, and designed or procedurally generated environments. VLN-CE and Interactive Gibson build on scanned homes, with the latter adding movable objects for interactive navigation (Krantz et al., 2020; Xia et al., 2020). GRUtopia and HumanoidVLN evaluate navigation in large-scale constructed or reconstructed environments (Wang et al., 2024; Pham et al., 2026), while BEHAVIOR-1K, RoboCasa365, HomeRobot, BiGym, HumanoidBench, and SIMPLE cover manipulation and whole-body control across curated or generated scenes (Li et al., 2023; Nasiriany et al., 2026; Yenamandra et al., 2023; Chernyadev et al., 2024; Sferrazza et al., 2024; Wei et al., 2026c). In contrast, LIVIN preserves the observed layouts, furniture, everyday belongings, and articulated fixtures of individual lived-in homes, retaining naturally occurring spatial constraints that jointly affect navigation and manipulation.

3 LIVIN Benchmark

LIVIN contains 30 digital twins of lived-in homes, comprising 319 rooms and 12808 specific object instances, including 1559 articulated objects. The collection covers a broad range of household scales, room functions, furnishing styles, and object compositions. We assess scenes in LIVIN along two dimensions: fidelity to the captured homes and preservation of real-world household complexity. Additional dataset statistics and qualitative examples are provided in Appendix A.

Table 1: **Comparison with existing indoor scene datasets.** Inst.-Spec. Mesh denotes instance-specific object meshes, and Artic. Struct. denotes modeled articulation. Obj. Dens. denotes object density (obj./m^2), and Artic. Obj. denotes the percentage of articulated objects. Plaus., Realism, and Mesh Qual. denote human preference rates for physical plausibility, scene realism, and mesh quality.

	Properties			Statistics			User Study		
Dataset	**Real-World Grounding**	**Inst.-Spec. Mesh**	**Artic. Struct.**	**Avg. Obj. / Room**	**Obj. Dens. (obj./m^2)**	**Artic. Obj. (%)**	**Plaus. ↑ (%)**	**Realism ↑ (%)**	**Mesh Qual. ↑ (%)**
3D-FRONT (Fu et al., 2021)	✗	✗	✗	6.93	0.39	–	15.33	10.10	12.60
ProcTHOR (Deitke et al., 2022)	✗	✗	✓	18.33	0.82	11.11	5.23	8.36	2.03
HSSD (Khanna et al., 2024)	✗	✗	✓	20.17	1.58	10.08	13.24	21.26	18.70
ReplicaCAD (Szot et al., 2021)	✓	✗	✓	30.48	0.38	5.38	10.80	4.18	3.66
MetaScenes (Yu et al., 2025)	✓	✗	✗	21.76	1.63	–	4.53	1.05	2.03
LIVIN	✓	✓	✓	**40.15**	**3.70**	**12.17**	**50.87**	**55.05**	**60.98**

Real-world correspondence. Each LIVIN scene is reconstructed from a specific lived-in home and grounded in its captured evidence. At the scene level, LIVIN preserves the architectural layout and fine-grained details of each home, including ceilings, niches, beams, and wall openings. At the object level, it provides instance-specific meshes and explicitly models the movable parts and kinematic relationships of articulated household objects. We compare LIVIN with other indoor scene datasets in Tab. 1 in terms of scene- and object-level properties. We further validate scene quality through a user study on physical plausibility, perceived realism, and reconstructed mesh quality. LIVIN receives consistently higher ratings across all three dimensions than the compared datasets, reflecting its high-fidelity reproduction of real-home structure, object arrangements details.

Reproducing household complexity. We further quantify how well LIVIN captures the object-level complexity of real household environments in Tab. 1. The results show that LIVIN achieves the highest object count and density among datasets with available statistics, while also containing a larger proportion of articulated objects than datasets reporting this metric. Together, these findings suggest that LIVIN better captures the object richness and interaction-relevant structure of lived-in household environments.

4 Replica Construction with Human Verification

Given the floorplan, posed multi-view images, and a metric scan of a real home, we develop a human-in-the-loop agentic pipeline for constructing its digital replica. We use GPT-6 Astra (OpenAI, 2026) as the reconstruction agent throughout the pipeline. The workflow consists of three stages: instance recognition, architectural reconstruction, and object generation and placement. Each stage is followed by human inspection and refinement to ensure consistency with the captured environment. Additional implementation details are provided in Appendix B.

4.1 Instance Recognition

We first establish a complete inventory of the objects present in each home to identify the instances to be reconstructed. The agent identifies and segments object instances across posed multi-view images, revisiting densely populated regions and consulting complementary views to recover objects that are partially or fully occluded in individual images. Cross-view detections are matched against the existing asset inventory to remove duplicates. For each instance, we record its semantic label, material attributes, associated room, and articulation properties. The articulation information determines how the object is generated in the next stage. The agent also selects a reference view of each object for subsequent generation, based on visibility, completeness, and image quality.

4.2 ARCHITECTURAL CONSTRUCTION

Reconstruction begins with the architectural structure of each home. Based on the 2D floorplan, metric scans, and available room metadata, we first build a coarse 3D architectural shell with the major walls, floors, ceilings, and openings. More details of the initial shell construction are provided in Appendix B.4. The agent then aligns this initial structure with the posed images, refines wall connections and openings, and adds observed architectural details such as niches, beams, and ceiling features. Windows are further reconstructed with frames and glass according to their observed configurations.

4.3 OBJECT GENERATION AND PLACEMENT

For object reconstruction and placement, we follow the generation routes determined during instance recognition and use different strategies for articulated and non-articulated objects.

Articulated objects. For articulated objects, the agent directly generates Blender code for in-context construction. It estimates each object's pose from the selected reference view and surrounding room geometry, while jointly modeling its geometry, materials, articulation, and placement. Coupling generation with placement allows the reconstructed geometry to conform to the observed footprint and surrounding structures. Functional components such as storage cavities and drawer boxes are also preserved, with movable parts explicitly represented through motion axes and kinematic relationships.

Non-articulated objects. For non-articulated objects, we use Rodin (Zhang et al., 2024) to generate the corresponding 3D assets. Given the selected reference view of each object, we use GPT-Image-2 (OpenAI, 2026) to generate a clean product-style image by removing occlusions and unrelated surroundings while preserving the object's appearance and constituent parts. Rodin then generates a high-fidelity mesh from this cleaned reference. The agent subsequently estimates each object's scale, pose, and placement from the constructed scene context and source observations. The placement process is required to preserve spatial relations such as stacking, containment, wall mounting, and suspension, while avoiding inter-object collisions to improve physical plausibility.

4.4 HUMAN REVIEW AND REFINEMENT

Automatic reconstruction may introduce errors that require human correction. We therefore review and refine the output of each construction stage before proceeding to the next, limiting error propagation. During instance recognition, reviewers correct detection errors, room assignments, and reference images for subsequent Rodin generation. For scene reconstruction, we develop an interactive review interface that enables reviewers to compare the reconstructed scene against the source observations and correct errors in geometry, articulation, appearance, and placement. For architectural elements and articulated objects, reviewers select the relevant structure or part and provide natural-language instructions to precisely guide the reconstruction agent toward the required corrections. Defective non-articulated objects can instead be regenerated with Rodin, while object poses and placements can be adjusted directly through the interface. Further details of the review and correction process are provided in Appendix C.

5 3D DETECTION AND RECONSTRUCTION

We consider two 3D tasks in the captured household environments: object detection and scene reconstruction. These tasks evaluate object recognition, localization, and geometry recovery under occlusion and dense object arrangements. Fig. 2 presents representative qualitative results for both tasks, comparing model predictions and reconstructions with the corresponding ground truth.

5.1 3D OBJECT DETECTION

We evaluate SceneScript (Avetisyan et al., 2024), EFM3D (Straub et al., 2024), SpatialLM (Mao et al., 2025), and Boxer (DeTone et al., 2026) on 268 rooms from the 30 homes in LIVIN using released checkpoints without fine-tuning. Method-specific inputs are constructed from ren-

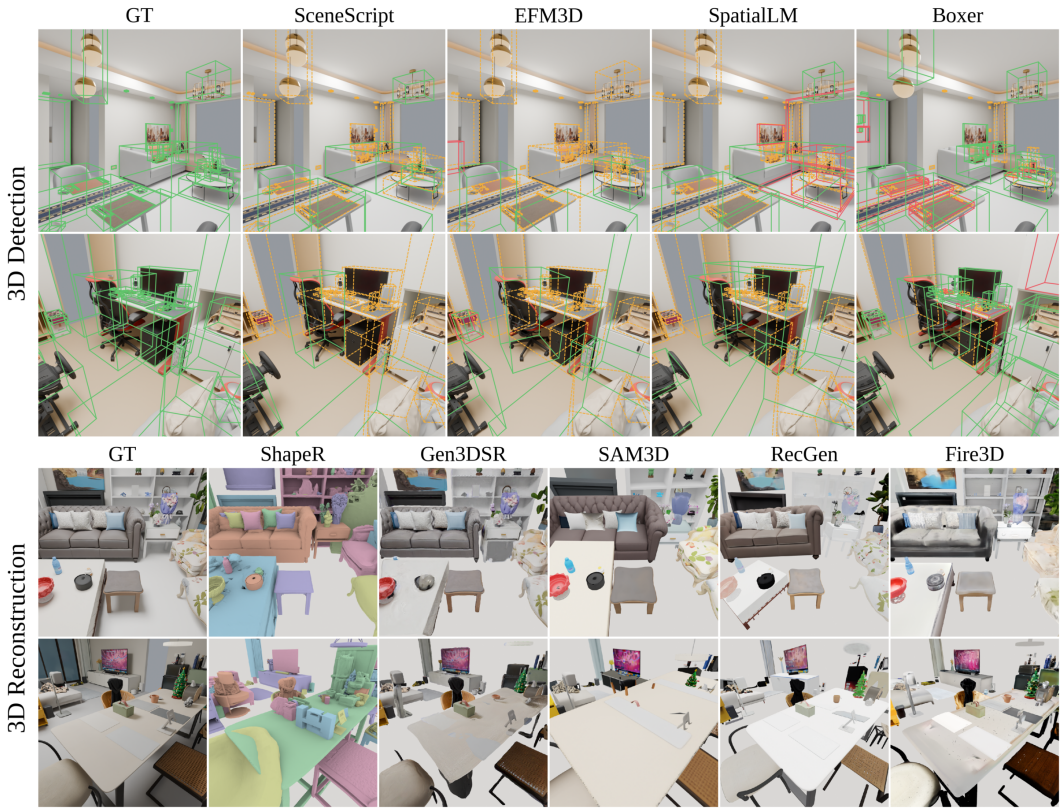


Figure 2: **Qualitative results for 3D detection and reconstruction.** Top: 3D detection predictions from SceneScript, EFM3D, SpatialLM, and Boxer compared with ground truth, where green boxes denote correct detections, yellow boxes indicate missed objects, and red boxes denote incorrect predictions. Bottom: 3D reconstructions from ShapeR, Gen3DSR, SAM3D, RecGen, and Fire3D compared with ground-truth.

Table 2: **Quantitative evaluation of 3D detection and reconstruction on LIVIN.** F-Score-S uses a surface-distance threshold of 0.05 m, and F1 uses a 3D box-matching threshold of 0.25.

3D Detection					
Method	**mAP ↑**	**Prec.@0.25/0.5 ↑**	**Recall@0.25/0.5 ↑**	**F1@0.25/0.5 ↑**	**3D IoU@0.25/0.5 ↑**
SceneScript (Avetisyan et al., 2024)	–	0.608 / 0.428	0.083 / 0.058	0.146 / 0.103	0.591 / 0.678
EFM3D (Straub et al., 2024)	0.071	0.617 / 0.362	0.096 / 0.056	0.166 / 0.097	0.541 / 0.657
SpatialLM (Mao et al., 2025)	–	**0.743 / 0.522**	0.111 / 0.078	0.192 / 0.135	**0.601 / 0.698**
Boxer (DeTone et al., 2026)	**0.306**	0.721 / 0.477	**0.417 / 0.276**	**0.528 / 0.349**	0.568 / 0.659

3D Reconstruction						
Method	**CD-S ↓**	**F-Score-S@0.05m ↑**	**F1@0.25 ↑**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
Gen3DSR (Ardelean et al., 2025)	0.214	0.336	0.627	14.03	0.678	0.369
SAM3D (Chen et al., 2026b)	0.133	0.381	0.685	12.07	0.625	0.460
RecGen (Zadaianchuk et al., 2026)	0.185	0.431	0.664	12.06	0.647	0.428
ShapeR (Siddiqui et al., 2026)	0.040	0.602	0.838	–	–	–
Fire3D (Xia et al., 2026a)	**0.022**	**0.732**	**0.880**	**14.88**	**0.701**	**0.360**

dered RGB-D observations: SceneScript and SpatialLM use point clouds, EFM3D uses multi-view RGB with semidense geometry and camera information, and Boxer uses RGB-D frames with OWLv2 (Minderer et al., 2023) 2D proposals. We evaluate class-agnostic room-level predictions against visible-object oriented boxes using Hungarian matching with 3D IoU, reporting precision, recall, F1, matched-box IoU. For methods that provide confidence scores, we additionally report



Figure 3: **Humanoid navigation in LIVIN.** LIVIN evaluates humanoid navigation in faithful replicas of real lived-in homes under naturally occurring spatial constraints. From left to right, examples show the robot turning into a room, moving sideways through a narrow passage, and stepping over a doorway threshold.

mAP averaged over IoU thresholds from 0.05 to 0.50 in increments of 0.05. As shown in Tab. 2, Boxer achieves the highest F1 but still limited recall, while SpatialLM attains the highest precision and matched-box IoU with only 0.111 recall at IoU 0.25. These results show that accurate localization of detected objects does not ensure complete scene coverage, and that object recovery remains a major challenge in LIVIN's object-dense household scenes.

5.2 3D Scene Reconstruction

For the 3D reconstruction task, we evaluate five methods on 128 views sampled from LIVIN scenes, with results reported in Tab. 2. All methods receive RGB, GT instance masks, depth, and camera calibration. Fire3D (Xia et al., 2026a) and ShapeR (Siddiqui et al., 2026) additionally use GT oriented bounding boxes, while ShapeR also takes depth-derived point clouds and object descriptions. ShapeR and Fire3D achieve higher recall and geometric accuracy overall, which may partly reflect the stronger priors provided by their additional GT inputs. SAM3D (Chen et al., 2026b), RecGen (Zadaianchuk et al., 2026), and Gen3DSR (Ardelean et al., 2025) show different trade-offs between geometry and appearance. SAM3D achieves lower surface error, while RecGen obtains a higher scene level F-Score. Gen3DSR is less accurate geometrically, with lower scene level F-Score and F1, but better preserves appearance, yielding the strongest PSNR, SSIM, and LPIPS among the three. Additional analysis and discussion are provided in Appendix D.

6 Navigation and Loco-Manipulation

We evaluate two embodied tasks in LIVIN: navigation and loco-manipulation. Unlike prior benchmarks built on purpose-designed or deliberately cluttered test environments, LIVIN provides faithful replicas of complete lived-in homes, preserving naturally occurring spatial constraints and object arrangements. This enables the evaluation of robot mobility and interaction under conditions that more closely reflect real household operations. Moreover, the diverse structures preserved in LIVIN naturally support a broad range of task settings, including narrow passages, interactive household fixtures and object manipulation across different room types.

6.1 Navigation

We formulate navigation as a long-horizon vision-language task for physically embodied humanoids, requiring the robot to follow route instructions using visual observations while executing feasible whole-body motion. Fig. 3 visualizes several representative navigation tasks in LIVIN. We compare current general-purpose multimodal models with state-of-the-art VLN methods, evaluating all methods without training or fine-tuning on LIVIN. Specifically, we consider GPT-6 Astra, OmniNav (Xue et al., 2026b), and DualVLN (Wei et al., 2026a) as the high-level navigation methods. All methods use GR00T-WBC (NVIDIA GEAR Team, 2025) as the controller and the same route instructions, while retaining their native observation interfaces: GPT-6 Astra uses on-demand head-camera RGB with interaction history, OmniNav uses three-view RGB with visual history, and DualVLN uses head-camera RGB with visual history.

Table 3: **Quantitative comparison of navigation methods on LIVIN.** SR denotes success rate and SPL denotes success weighted by path length; both are reported in percent. Collision Count is the mean number of robot–environment collision events, and NE denotes the mean geodesic distance from the robot's final position to the goal in meters.

Method	SR (%) ↑	SPL (%) ↑	Collision Count ↓	NE (m) ↓
OmniNav (Xue et al., 2026b)	6.67	3.66	80.59	6.69
DualVLN (Wei et al., 2026a)	4.44	2.56	**42.11**	5.97
GPT-6 Astra (OpenAI, 2026)	**41.48**	**23.59**	55.41	**2.16**

Our experiments cover 45 instruction-following routes across 30 LIVIN scenes using a Unitree G1 humanoid, with MuJoCo (Todorov et al., 2012) for physics simulation and Isaac Sim (NVIDIA, 2026) for RGB rendering. Results are reported in Tab. 3. GPT-6 Astra performs best overall, achieving a 41.48% success rate, the highest SPL, and the lowest navigation error. This may stem from Astra's stronger scene reasoning and information integration: in most trials, the robot maintains the correct motion direction and successfully selects the instructed branch at route intersections. The two specialized VLN baselines exhibit different failure modes. We find that OmniNav sometimes gets stuck in simple collision regions, repeatedly attempting similar motions without escaping, which leads to the highest collision count among the evaluated methods. In contrast, DualVLN recovers more effectively from such collisions but more often drifts from the intended route, resulting in low success and high navigation error. GPT-6 Astra is less affected by these issues, but can still fail to accurately reach the intended region because of imperfect scene understanding in complex household environments, or become limited by local motion planning in tightly constrained spaces. These results expose remaining challenges in current navigation systems under realistic household conditions. Detailed implementation, experimental settings, and further discussion are provided in Appendix E.

6.2 Loco-Manipulation

We evaluate instruction-conditioned household loco-manipulation, requiring a humanoid to coordinate locomotion and object interaction under the spatial constraints of LIVIN's lived-in homes. Tasks span four families: **object grasping and placement**, **object pushing and pulling**, **articulated-object manipulation**, and **long-horizon tasks** with dependent subgoals. Examples include grasping and placing objects in constrained spaces, moving obstructions to access targets or clear passages, and opening a cabinet before placing an object inside (Fig. 4). Household arrangements therefore form part of the task rather than merely its background: robots must operate within existing spatial constraints and, when necessary, modify the environment to enable subsequent locomotion or manipulation. We compare two multimodal large language models (MLLMs), DeepSeek-V4.1-Flash (DeepSeek-AI, 2026) and GPT-6 Astra (OpenAI, 2026), with the humanoid foundation model Ψ_0 (Wei et al., 2026b), evaluating all methods without training or fine-tuning on LIVIN (Tab. 4).

We evaluate 35 tasks across 30 LIVIN homes using a Unitree G1 humanoid with Dex3 hands and the physics and rendering backends described in Sec. 6.1. Both MLLMs receive task instructions, head-camera RGB images, and robot state information, and control the robot through a shared SONIC-based whole-body controller (Luo et al., 2026). Ψ_0 uses its released task experts and native controllers. Each method is evaluated in three trials per task from the same initial configuration. We report success rate (SR) across all task families and mean process score (PS) for long-horizon tasks. Further experimental details are provided in Appendix F.

Results and analysis. As shown in Tab. 4, GPT-6 Astra achieves the highest success rates across all four task families, while DeepSeek-V4.1-Flash achieves limited success and Ψ_0 completes no tasks under the evaluated zero-shot configuration. Qualitative observations reveal target-localization errors and unsuccessful recovery attempts for DeepSeek-V4.1-Flash. For GPT-6 Astra, a cup-grasping example illustrates difficulties in transitioning from target approach to manipulation: the robot approaches the target but makes unintended contact during reaching before establishing a stable grasp. Performance also varies across interaction types, with GPT-6 Astra achieving 61.9% SR



Figure 4: **Loco-manipulation tasks.** Examples of object grasping and placement, object pushing and pulling, articulated-object manipulation, and long-horizon tasks in household environments.

Table 4: **Loco-manipulation performance across task families on LIVIN.** SR denotes end-to-end success rate in percent and is reported for all four task families. PS denotes the mean process score across trials (0–100) and is reported only for long-horizon tasks. Higher is better.

Method	Grasping and placement	Pushing and pulling	Articulated-object manipulation	Long-horizon tasks	
	SR (%) ↑	SR (%) ↑	SR (%) ↑	SR (%) ↑	PS ↑
DeepSeek-V4.1-Flash (DeepSeek-AI, 2026)	6.7	20.0	19.0	0.0	2.8
GPT-6 Astra (OpenAI, 2026)	**24.4**	**33.3**	**61.9**	**12.5**	**15.9**
Ψ_0 (Wei et al., 2026b)	0.0	0.0	0.0	0.0	0.0

on articulated-object manipulation but only 24.4% on grasping and placement. Long-horizon completion remains limited: GPT-6 Astra achieves 12.5% SR and a mean PS of 15.9, while DeepSeek obtains a mean PS of 2.8 with no successful completions, indicating limited progress on intermediate physical subgoals. Together, these results expose limitations beyond target approach, in both establishing effective physical contact and completing dependent interactions within household tasks.

7 Conclusion

We presented LIVIN, a benchmark of digital twins of lived-in homes for spatial and embodied intelligence. By preserving real-world scene structure, object arrangements and articulation, LIVIN enables evaluation across 3D perception, reconstruction, navigation, and loco-manipulation in realistic household environments. Our results show that current methods still struggle with dense object arrangements, frequent occlusion, and physically constrained interactions. By exposing these challenges in realistic household environments, LIVIN provides a testbed for driving progress toward more robust and generalizable embodied systems.

AI USE STATEMENT

We used ChatGPT and Codex (OpenAI) to assist with literature search, manuscript drafting and revision, LaTeX editing, and discussions of task design, evaluation protocols. Generative AI also forms part of our research workflow: GPT-6 Astra supports instance recognition, Blender-based scene construction, and initial physical-parameter estimation; GPT-Image-2 prepares object reference images, and Rodin generates 3D assets. These uses and the human review and correction procedures are described in Sec. 4 and Appendices B, C, and F. The authors take full responsibility for the content, methods, results, and conclusions of this paper.

REFERENCES

Andreea Ardelean, Mert Özer, and Bernhard Egger. Generalizable 3d scene reconstruction via divide and conquer from a single view. In *International Conference on 3D Vision (3DV)*, 2025.

Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Niessner. Scan2cad: Learning cad model alignment in rgb-d scans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, Jakob Engel, Edward Miller, Richard Newcombe, and Vasileios Balntas. Scenescript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision (ECCV)*, 2024.

Junhao Chen, Xinghao Chen, Henghaofan Zhang, Zihao Qiao, Saining Zhang, Yongzhi Li, Ruqi Huang, Sisi Li, Yimin Sheng, Jianyi Zhu, and Hao Zhao. Engine-native editable 3D world reconstruction with objects and lighting. *arXiv preprint arXiv:2607.20889*, 2026a.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 2026b.

Yongwei Chen, Yushi Lan, Shangchen Zhou, Tengfei Wang, and Xingang Pan. Sar3d: Autoregressive 3d object generation and understanding via multi-scale 3d vqvae. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28371–28382. IEEE, 2025.

Nikita Chernyadev, Nicholas Backshall, Xiao Ma, Yunfan Lu, Younggyo Seo, and Stephen James. Bigym: A demo-driven mobile bi-manual manipulation benchmark. *arXiv preprint arXiv:2407.07788*, 2024.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. *arXiv preprint arXiv:2609.19969*, 2026.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in neural information processing systems*, 35:5982–5994, 2022.

Daniel DeTone, Tianwei Shen, Fan Zhang, Lingni Ma, Julian Straub, Richard Newcombe, and Jakob Engel. Boxer: Robust lifting of open-world 2d bounding boxes to 3d. In *European Conference on Computer Vision (ECCV)*, 2026.

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10913–10922. IEEE, 2021.

Daoyi Gao, Yawar Siddiqui, Lei Li, and Angela Dai. Meshart: Generating articulated meshes with structure-guided transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 618–627. IEEE, 2025.

Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. Midi: Multi-instance diffusion for single image to 3d scene generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23646–23657. IEEE, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16384–16393. IEEE, 2024.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120. Springer, 2020.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pp. 80–93. PMLR, 2023.

Jiayi Liu, Hou In Ivan Tam, Ali Mahdavi-Amiri, and Manolis Savva. Cage: Controllable articulation generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17880–17889. IEEE, 2024.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117):eaed4592, 2026.

Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. Spatiallm: Training large language models for structured indoor modeling. In *Advances in Neural Information Processing Systems*, 2025.

Yanxu Meng, Haoning Wu, Ya Zhang, and Weidi Xie. Scenegen: Single-image 3d scene generation in one feedforward pass. In *2026 International Conference on 3D Vision (3DV)*, pp. 543–553. IEEE, 2026.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations (ICLR)*, 2026.

NVIDIA. NVIDIA Isaac Sim: Robotics simulation and synthetic data generation. `https://developer.nvidia.com/isaac/sim`, 2026.

NVIDIA GEAR Team. GR00T-WholeBodyControl: Whole-body control for humanoid robots. `https://github.com/NVlabs/GR00T-WholeBodyControl`, 2025.

OpenAI. GPT-6 Astra: A New Generation of Intelligence, 2026. URL `https://openai.com/index/gpt-6-astra/`.

OpenAI. GPT-Image-2: Image generation model, 2026. URL `https://openai.com/index/gpt-image-2/`.

Nicholas Pfaff, Thomas Cohn, Sergey Zakharov, Rick Cory, and Russ Tedrake. Scenesmith: Agentic generation of simulation-ready indoor scenes. In *Forty-third International Conference on Machine Learning*, 2026.

Quan-Dung Pham, Anh Dao, The-Anh Nguyen, Minh Nguyen-Dinh, Phuong Nam Dang, Tri Pham, Hung Tran, Bach Dao, Tuyen P Le, Truong Nguyen, et al. Humanoidvln: A physics-grounded simulator and benchmark for vision-language navigation across diverse humanoid embodiments. *arXiv preprint arXiv:2608.12860*, 2026.

Minghan Qin, Yuang Wang, Xiuyu Yang, Yushi Long, Yujian Zhang, Ruihuan Wang, Kai Ye, Yangang Zhang, and Hang Li. Lucida: Parse, generate, and place for composable real-to-sim scene modeling. *arXiv preprint arXiv:2608.30821*, 2026.

Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21783–21794, June 2024.

Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv preprint arXiv:2403.10506*, 2024.

Yawar Siddiqui, Duncan Frost, Samir Aroudj, Armen Avetisyan, Henry Howard-Jenkins, Daniel DeTone, Pierre Moulon, Qirui Wu, Zhengqin Li, Julian Straub, Richard Newcombe, and Jakob Engel. Shaper: Robust conditional 3d shape generation from casual captures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27157–27168, June 2026.

Julian Straub, Daniel DeTone, Tianwei Shen, Nan Yang, Chris Sweeney, and Richard Newcombe. Efm3d: A benchmark for measuring progress towards 3d egocentric foundation models. *arXiv preprint arXiv:2406.10224*, 2024.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266, 2021.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Hanqing Wang, Jiahe Chen, Wensi Huang, Qingwei Ben, Tai Wang, Boyu Mi, Tao Huang, Siheng Zhao, Yilun Chen, Sizhe Yang, Peizhou Cao, Wenye Yu, Zichao Ye, Jialun Li, Junfeng Long, Zirui Wang, Huiling Wang, Ying Zhao, Zhongying Tu, Yu Qiao, Dahua Lin, and Jiangmiao Pang. GRUtopia: Dream general robots in a city at scale. *arXiv preprint arXiv:2407.10943*, 2024.

Penghao Wang, Siyuan Xie, Hongyu Yan, Xianghui Yang, Jingwei Huang, Chunchao Guo, and Jiayuan Gu. Artllm: Generating articulated assets via 3d llm. *arXiv preprint arXiv:2603.01142*, 2026.

Meng Wei, Chenyang Wan, Peng Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, et al. Ground slow, move fast: A dual-system foundation model for generalizable vision-language navigation. In *International Conference on Learning Representations*, volume 2026, pp. 12380–12396, 2026a.

Songlin Wei, Hongyi Jing, Boqian Li, Zhenyu Zhao, Jiageng Mao, Zhenhao Ni, Sicheng He, Jie Liu, Xiawei Liu, Kaidi Kang, et al. Ψ_0: An open foundation model towards universal humanoid loco-manipulation. *arXiv preprint arXiv:2603.12263*, 2026b.

Songlin Wei, Zhenhao Ni, Jie Liu, Zhenyu Zhao, Junjie Ye, Hongyi Jing, Junkai Xia, Xiawei Liu, Michael Leong, Liang Heng, et al. Simple: Simulation-based policy learning and evaluation for humanoid loco-manipulation. *arXiv preprint arXiv:2606.08278*, 2026c.

Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41(4):1–18, 2022.

Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 5(2): 713–720, 2020.

Hongchi Xia, Tianhang Cheng, Wei-Chiu Ma, and Shenlong Wang. FIRE3D: Feed-forward interactive 3d scene reconstruction within a minute. *arXiv preprint arXiv:2609.08848*, 2026a.

Hongchi Xia, Xuan Li, Zhaoshuo Li, Qianli Ma, Jiashu Xu, Ming-Yu Liu, Yin Cui, Tsung-Yi Lin, Wei-Chiu Ma, Shenlong Wang, et al. Sage: Scalable agentic 3d scene generation for embodied ai. *arXiv preprint arXiv:2602.10116*, 2026b.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, et al. Native and compact structured latents for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14419–14429, 2026.

Han Xue, Sikai Liang, Zhikai Zhang, Zicheng Zeng, Yun Liu, Yunrui Lian, Jilong Wang, Qingtao Liu, Xuesong Shi, and Li Yi. Collision-free humanoid traversal in cluttered indoor scenes. *IEEE Robotics and Automation Letters*, 11(8):9183–9190, 2026a. doi: 10.1109/LRA.2026.3703590.

Xinda Xue, Junjun Hu, Minghua Luo, Shichao Xie, Jintao Chen, Zixun Xie, Quan Kuichen, Wei Guo, Zedong Chu, Mu Xu, et al. Omninav: A unified framework for prospective exploration and visual-language navigation. In *International Conference on Learning Representations*, volume 2026, pp. 106216–106231, 2026b.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16277–16287. IEEE, 2024.

Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Lan Xu, Wei Yang, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image. *ACM Transactions on Graphics (TOG)*, 44(4):1–19, 2025.

Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John Turner, et al. Homerobot: Open-vocabulary mobile manipulation. *arXiv preprint arXiv:2306.11565*, 2023.

Huangyue Yu, Baoxiong Jia, Yixin Chen, Yandan Yang, Puhao Li, Rongpeng Su, Jiaxin Li, Qing Li, Wei Liang, Zhu Song-Chun, Tengyu Liu, and Siyuan Huang. Metascenes: Towards automated replica creation for real-world 3d scans. In *Conference on Computer Vision and Pattern Recognition(CVPR)*, 2025.

Andrii Zadaianchuk, Leonardo Barcellona, Lennard Schuenemann, Christian Gumbsch, Zehao Wang, Muhammad Zubair Irshad, Fabien Despinoy, Rahaf Aljundi, Stratis Gavves, and Sergey Zakharov. Reconstruction by generation: 3d multi-object scene reconstruction from sparse observations. In *European Conference on Computer Vision*, pp. 498–519. Springer, 2026.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions On Graphics (TOG)*, 43(4):1–20, 2024.

Weipeng Zhong, Peizhou Cao, Yichen Jin, Luo Li, Wenzhe Cai, Jingli Lin, Hanqing Wang, Zhaoyang Lyu, Tai Wang, Xudong Xu, et al. Internscenes: A large-scale simulatable indoor scene dataset with realistic layouts. *Advances in Neural Information Processing Systems*, 38, 2026.

Matt Zhou, Ruining Li, Xiaoyang Lyu, Zhaomou Song, Zhening Huang, Chuanxia Zheng, Christian Rupprecht, Andrea Vedaldi, and Shangzhe Wu. Articraft: An agentic system for scalable articulated 3d asset generation. *arXiv preprint arXiv:2605.15187*, 2026.

A Dataset Statistics

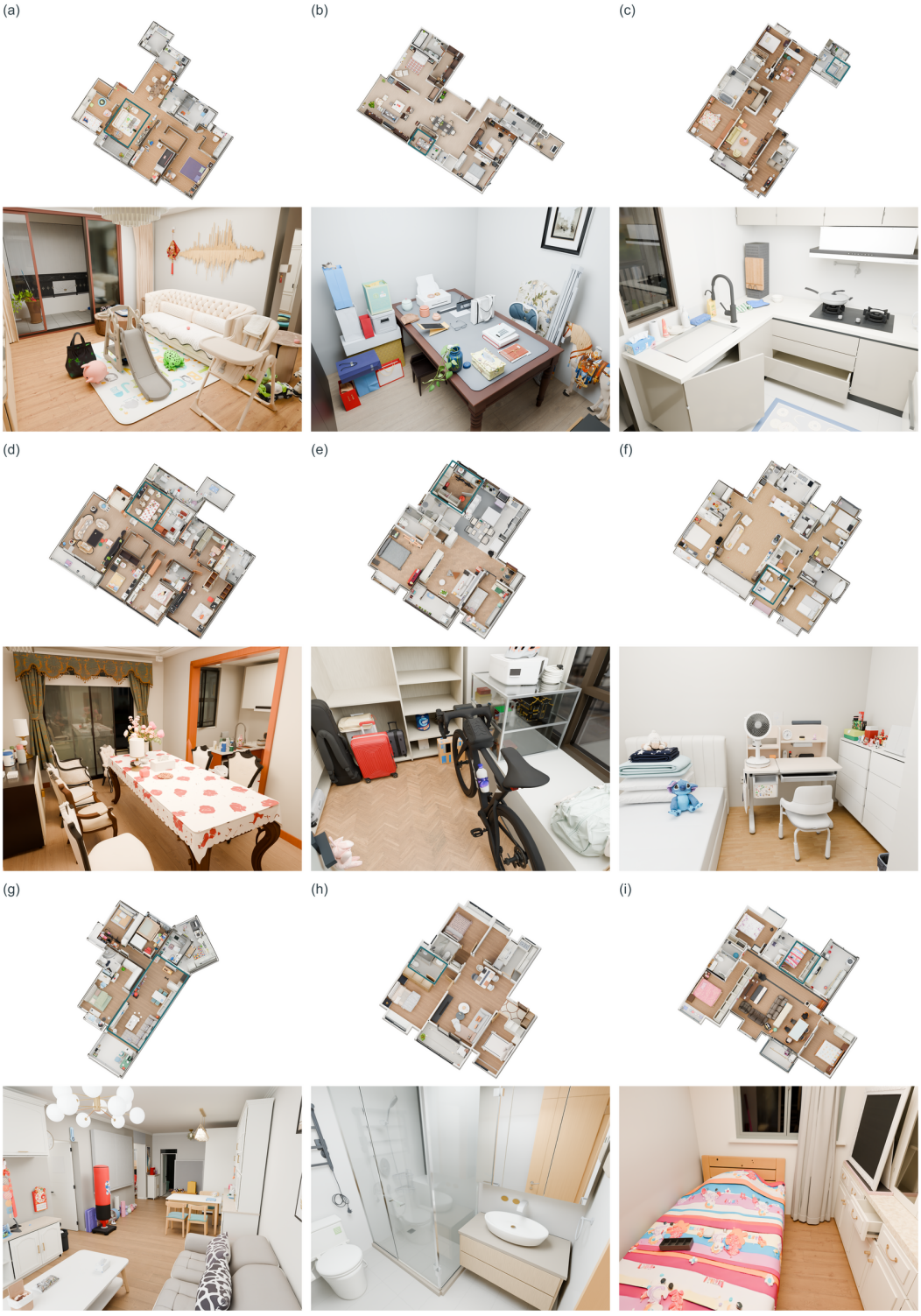


Figure 5: **Gallery.** Top down view and zoom in view.

We further characterize the LIVIN benchmark through additional dataset statistics and qualitative visualizations. Fig. 5 shows top-down and close-up views of the reconstructed homes, illustrating

their layouts and object arrangements. These homes span a diverse set of room types, as shown in Fig. 7a, with bedrooms and bathrooms being the most common, followed by balconies, kitchens, and living rooms. We further report the mean number of articulated objects per room type in Fig. 7b; living rooms and kitchens contain the highest averages, with 9.0 and 7.7 articulated objects per room, respectively, while bedrooms and bathrooms also exhibit substantial articulated content. Representative articulated assets and their movable configurations are visualized in Fig. 6. Together, these statistics and visualizations characterize the diversity, object density, real-world correspondence, and interaction-relevant structure of LIVIN.

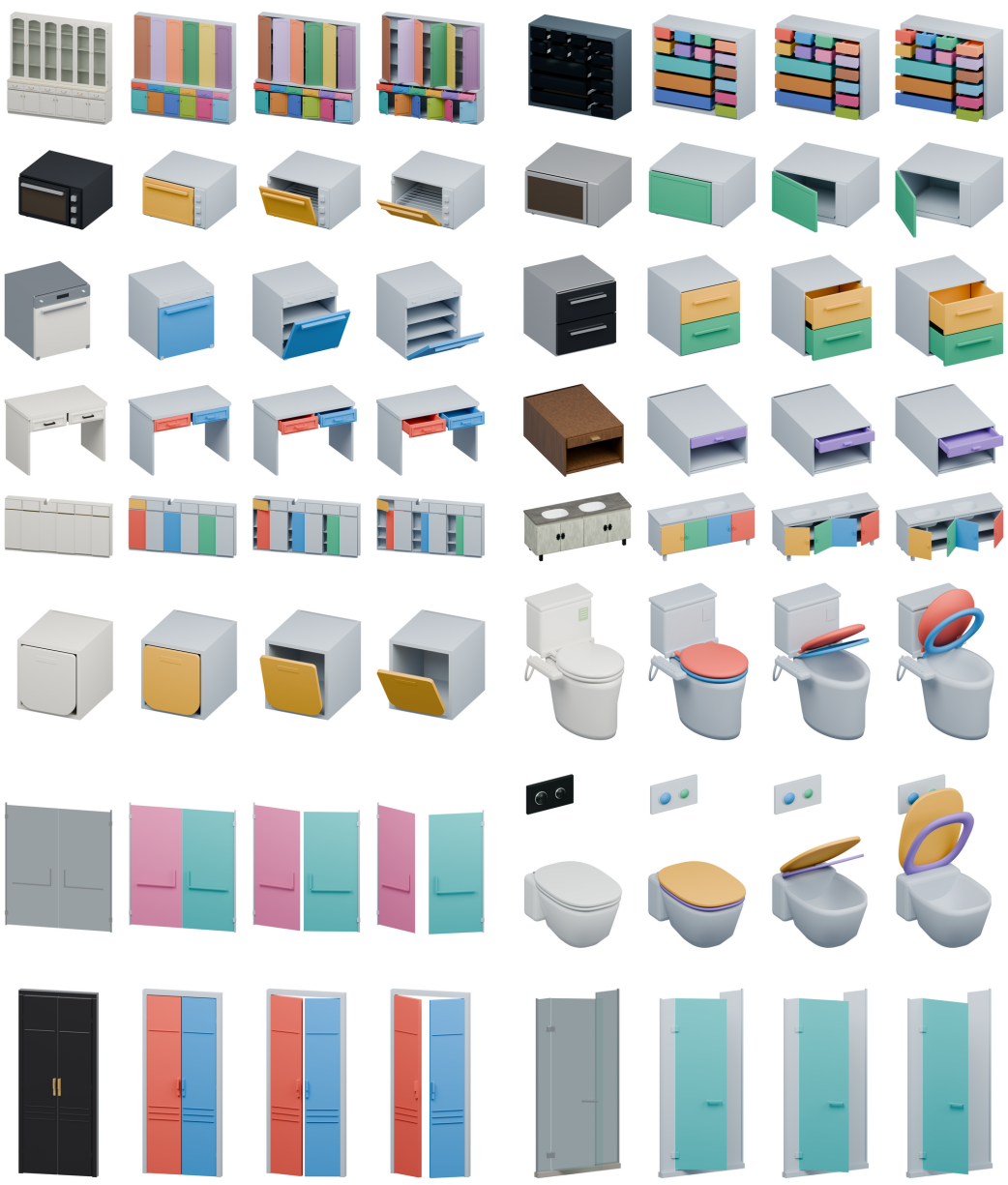
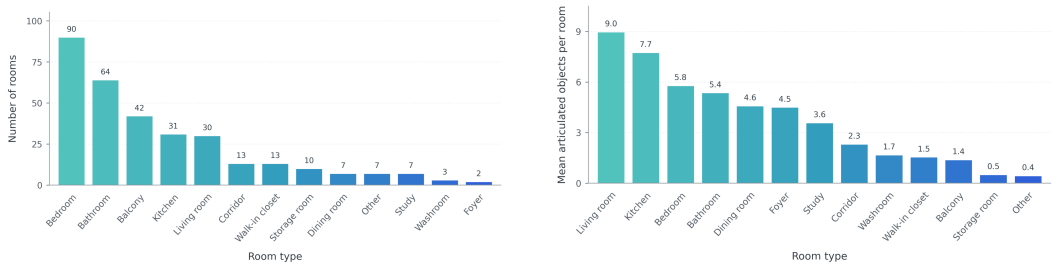


Figure 6: **Articulated objects in LIVIN**. Each quartet shows original appearance and motion frames. Colors identify moving assemblies consistently within each object; gray denotes stationary structure.



(a) Room-type distribution.

(b) Articulated objects across room types.

Figure 7: **Statistics of the LIVIN scene collection.** We report (a) the distribution of room types and (b) the distribution of articulated objects across room types.

B Automated Annotation and Scene Construction

B.1 Input Data and Spatial References

Each home is reconstructed from three complementary inputs, illustrated in Fig. 8: a floorplan, a metric scan, and posed multi-view images. The floorplan provides room organization and an initial architectural layout, while the metric scan supplies coarse 3D geometry and scale. The posed images provide the detailed visual evidence needed to identify individual objects, recover their appearance, and infer their spatial arrangements. We align these inputs in a common coordinate system before reconstruction. The posed images retain camera calibration and projection information, allowing visual observations to be related to the reconstructed geometry. We also have some structural metadata, including room boundaries and semantic labels, wall geometry and heights, door and window locations and dimensions, and room connectivity, which serve as spatial constraints for constructing the initial architectural structure.



(a) Floor plan

(b) Metric Scans

(c) Posed images

Figure 8: **Input overview.** Each home is represented by a 2D floorplan, a metric scan, and posed multi-view images.

B.2 Scene-to-Assets Annotation

We create a skill to guide the agent through instance identification, cross-view association, and evidence preparation. Dense or occluded regions are examined using enlarged crops or complementary views. Appearance, landmarks, and spatial evidence are used to match observations with existing entries and remove duplicate instances.

Table 5: **Rodin parameters used for object generation.**

Parameter	Setting
Generator Tier	Gen-2.5-High
Mesh Mode	Raw
Quality	Low
Output	GLB
Material	PBR
Texture Mode	High
Texture Delight	Enabled
HD Texture	Disabled
Geometry Instruction Mode	Faithful

Semantic and articulation annotations. We categorize each instance as articulated or non-articulated, and as large furniture or a small object. The furniture/small-object label provides the agent with a placement-order reference in the later reconstruction stage. Separate fields record semantic labels, rigid/soft/partially soft composition, and articulation evidence. Each judgment is assigned a confidence level and marked as observed, inferred, or unresolved.

Reference selection and product images. The agent selects a clear, sufficiently complete captured view and any needed auxiliary observations while continuing coverage inspection. For non-articulated small objects, the captured view is converted into a product-style reference image using GPT-Image-2 (OpenAI, 2026). The current package requests GPT-Image-2 at medium quality and 1024×1024, retaining the original response and delivered image separately. Generated hidden surfaces remain inferences and are checked against capture evidence.

B.3 Rodin Generation

Rodin uses the reference images described above to generate non-articulated 3D assets. Tab. 5 lists the current integration settings. During subsequent human-guided replacement and refinement, the image hash, prompt, generation parameters, and generation identifier are recorded for each regeneration job, allowing replacement assets to be traced throughout the correction process.

B.4 Blender-based Scene Generation

We use a coordinating skill to assign room-level tasks to the reconstruction agent, specifying the active stage, input scene, source inventory, accepted base, coordinate conventions, and expected outputs. Tab. 6 summarizes these responsibilities. To improve efficiency and reduce generation cost, the agent performs a single feed-forward pass at each stage without iterative self-refinement. Remaining errors are then identified by human reviewers, who provide precise, localized correction instructions to avoid repeated or unnecessary agent revisions.

Initial Structure Generation. We obtain the floor elevation from the room metadata and construct an initial 3D architectural structure using the corresponding wall heights and thicknesses. Door and window openings are reserved according to their recorded locations and dimensions. We then align this coarse structure with the metric scan and refine its geometry accordingly. In particular, when the reconstructed ceiling is substantially higher than the ceiling observed in the scan, we lower it and truncate the surrounding walls to the revised height for better geometric consistency.

Modeling in context. Articulated objects are generated jointly with their placement so that geometry can adapt to the surrounding scene rather than being positioned only after modeling. For example, cabinets and counters are constructed to fit supporting walls, available footprints, and neighboring structures. The agent also models internal compartments, drawer boxes, handles, and separate fixed and movable parts, with rotation axes, pivots, and sliding directions specified according to the inferred mechanism. To facilitate human verification, we also generate a simple animation for each articulated object to visualize its motion.

Table 6: **Responsibilities of the construction skills.** Each skill specifies task instructions and required deliverables for scene construction.

Skill	Guidance for the Generated Blender Scene
Coordination	Assign stage-specific room tasks, preserve source identities and accepted bases, and collect complete room outputs before publication.
Structure	Import the initial shell; check cameras and scale; refine walls, openings, recesses, and visible architectural details; reconstruct windows at the reference pose.
Structure-Aware Fitting and Articulation	Jointly model object geometry, materials, placement, internal structures, and movable parts in the surrounding room context; record articulation mechanisms, support relations, and observed or inferred parameters.
Object Placement	Import assigned meshes, preserve their parts and materials, place supports before supported objects, ensure collision-free placement, and record root transforms and support relations.

B.5 Construction Ablations

We ablate the construction and correction strategy using five settings, with some qualitative results shown in Fig. 9. **Code-Only** reconstructs the entire scene directly from floorplans, posed images, and metric scans using Blender code, followed by up to five rounds of agent self-refinement. **No Refinement**, **Single-Round Refinement**, and **Iterative Refinement** use the same staged initialization with fixed coarse structural meshes, detection results, and Rodin assets, but apply zero, one, and up to three rounds of autonomous refinement, respectively. To assess the efficiency of the reconstruction process, we also record token consumption. **Single-Round Refinement** and **Iterative Refinement** each require over 100 million tokens, compared with just over 70 million for our workflow. The results show that our workflow improves physical plausibility and mesh quality while reducing inter-object penetrations. These gains arise from both our generation strategies and human-in-the-loop verification. Blender-based construction preserves articulated structure and kinematics, while Rodin provides high-fidelity geometry for non-articulated objects. Human review further improves geometry and placement accuracy, while avoiding redundant agent-side checking and ineffective iterative refinement, leading to a better balance between reconstruction quality and computational cost.

C Human Review and Verification

C.1 Review and Correction Workflow

Before scene reconstruction, reviewers inspect the instance inventory against the source observations, with particular attention to low-confidence or unresolved cases from the previous stage. They resolve uncertain annotations, add missing instances, remove duplicates, and revise semantic labels or room assignments when needed. Reviewers can also update the reference image used for downstream asset generation. When necessary, they provide a text prompt to GPT-Image-2 to regenerate a cleaner product-style reference while preserving the target object's identity and appearance.

For 3D reconstruction, reviewers first inspect and correct the generated architectural structure. Object generation and placement are then performed automatically, after which the resulting object-level reconstruction is reviewed in a second stage. Articulated and non-articulated objects are corrected using different strategies, as summarized in Tab. 7.

Architectural structures and articulated objects permit at most two agent repair rounds based on human feedback. Each round starts from a copy of the saved human-adjusted Blender scene, addresses the latest review feedback, and returns all rooms for another round of inspection. Non-articulated objects are corrected through Rodin regeneration or manual pose adjustment, without additional agent repair rounds.



Figure 9: **Construction ablation.** Qualitative comparison of Code-Only, Single-Round Refinement, Iterative Refinement, No Refinement, and our human-guided construction strategy.

Table 7: **3D reconstruction review process.** Each phase is reviewed for its corresponding reconstruction aspects, followed by targeted correction and completion.

Phase	Inspection Aspects	Correction and Completion
Structure	Room boundaries, wall connections, openings, scale, and alignment with source observations.	Select the relevant structural element and provide localized correction feedback. Each room is explicitly confirmed before proceeding.
Articulated Objects	Geometry, dimensions, materials, placement, structural fit, and articulation.	Select the relevant object or part for correction, adjust its pose, and resolve inter-object penetration.
Non-Articulated Objects	Appearance, geometry, placement, support, and relations to surrounding objects.	Adjust object poses, resolve inter-object penetration, or regenerate defective assets.

C.2 Reconstruction Review Interface

Model–observation comparison. The interface places the reconstructed model beside the captured posed images, as shown in Fig. 10. Selecting an asset initializes the best captured view at its representative observation point and viewing direction. Reviewers can switch among posed images and zoom in for closer inspection.

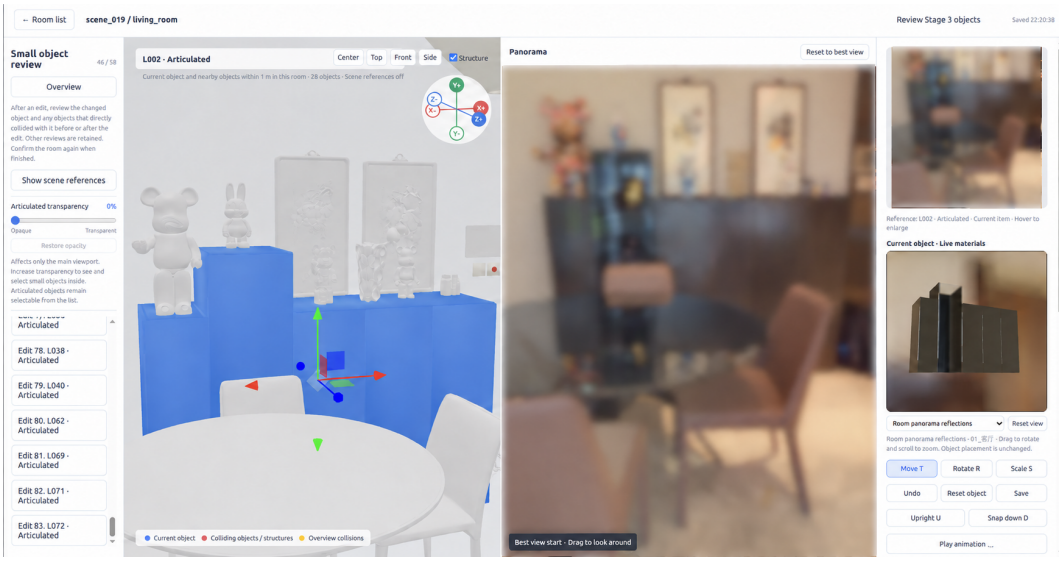


Figure 10: **Snapshot of the human review interface.** The reconstructed scene is shown on the left, the corresponding posed reference image on the right, and an additional panel on the far right displays object material information for inspection.

Geometric context and collision feedback. Scenes in LIVIN contain many high-resolution object meshes, making it memory-intensive to load the entire scene into the review interface at once. We therefore display the selected object together with nearby objects within 1 m, while always including its detected collision partners. The selected object is highlighted in blue and collision partners in red, allowing reviewers to inspect local geometric relationships without loading all scene assets simultaneously. Collision feedback is computed on reference-pose meshes using bounding-box filtering followed by BVH triangle intersection.

Pose editing and snapping. For placement errors that do not require asset regeneration, reviewers can directly correct object poses through translation, rotation and uniform scaling. However, for floating objects or objects that must closely align with supporting surfaces, manual pose adjustment alone cannot reliably establish precise contact. We therefore introduce axis-based snapping, which translates an object along a user-specified world axis while preserving its rotation and scale. Candidate contact surfaces are first filtered using bounding boxes, after which BVH-accelerated mesh intersection tests search for the first contact and a binary search refines the contact position. The final position is offset by 2 mm from the detected contact surface along the snapping direction, providing a small safety margin against mesh penetration. This allows supported objects to settle onto floors or furniture and wall-mounted or switch-like structures to align closely with architectural surfaces.

Animation and material inspection. To better visualize the articulation of articulated objects, the interface allows reviewers to play simple motion previews of their movable parts. Rigid-part animation previews expose the authored motion relative to the reference geometry and retain explicit statuses for absent or unsupported motion. As shown in the far-right panel of Fig. 10, an on-demand PBR preview allows reviewers to inspect object textures, transparency, reflections, and material channels under controlled illumination.

Table 8: **Output coverage and reconstruction accuracy on the clutter-like proxy subset of LIVIN.** Output coverage reports the number and percentage of generated objects, while 3D box recall and $\overline{F}_{\text{obj}}$ evaluate object-level reconstruction accuracy.

Method	Output Coverage ↑	3D Box Recall ↑	$\overline{F}_{\text{obj}}$ ↑
Gen3DSR (Ardelean et al., 2025)	132/306 (43.14%)	34.97%	0.2586
RecGen (Zadaianchuk et al., 2026)	**306/306 (100.00%)**	42.48%	0.3558
SAM3D (Chen et al., 2026b)	**306/306 (100.00%)**	53.59%	0.3686
Fire3D (Xia et al., 2026a)	292/306 (95.42%)	72.55%	0.5774
ShapeR (Siddiqui et al., 2026)	**306/306 (100.00%)**	**85.95%**	**0.7510**

Text feedback and bounding-box control. The interface provides a dedicated text box whose purpose depends on the selected element: reviewers enter correction instructions for the GPT agent when reviewing room structures or articulated objects, and generation prompts for Rodin when reviewing non-articulated objects. For Rodin generation, reviewers also can optionally adjust a 3D bounding box through axis dragging or numerical inputs to control the object's width, height, and depth proportions. Bounding-box control helps Rodin reconstruct object geometry more accurately by constraining its overall proportions.

D 3D Scene Reconstruction

Main evaluation protocol. CD-S measures the symmetric mean squared nearest-neighbor surface distance, in m^2. The main-table F-Score-S uses a fixed distance threshold of $0.05\,\mathrm{m}$, while object-level F1 uses a 3D box-matching threshold of 0.25. F1 is computed from TP, FP, and FN pooled across views; the remaining main-table metrics are averaged over views.

For appearance evaluation, predictions are independently rendered using common cameras, lighting, and black backgrounds, without GT-mask clipping. GT images retain brightness-adjusted RGB values within the selected target masks and are black elsewhere. PSNR, SSIM, and LPIPS are computed over the full images. ShapeR is excluded from appearance evaluation because it produces untextured geometry.

Clutter-focused evaluation. We further compare different reconstruction methods in terms of small-object coverage and geometric fidelity. A common evaluation subset is drawn from the main 128-view LIVIN test set for all methods. We first use GT name filtering to select clutter-related objects, and then apply physical and projected size constraints to select small targets. A minimum projected-size threshold further removes barely visible instances. The resulting subset contains 306 object–view records covering 287 distinct objects across 76 views. Fig. 11 shows examples of the selected objects.

We report output coverage, 3D box recall, and aggregate surface F-score in Tab. 8. ShapeR performs best on this subset, achieving full output coverage together with substantially higher box recall and surface F-score. Fire3D also achieves strong geometric accuracy, but its lower coverage indicates that some objects are not reconstructed at all. Gen3DSR exhibits the opposite limitation: its small-object filtering leaves more than half of the selected records without output, making coverage the dominant bottleneck. Most methods show a marked drop in 3D box recall under cluttered small-object conditions, and all methods still leave substantial room for improvement. These results demonstrate that LIVIN effectively exposes current limitations in small-object reconstruction and provides a meaningful benchmark for comparing reconstruction methods in challenging cluttered environments.

E Navigation Evaluation Details

E.1 Implementation Details

All GR00T-based systems use the same whole-body control backend. GPT-6 Astra (OpenAI, 2026), configured with high reasoning effort, observes RGB images from the mounted head camera on



Figure 11: **Selected small clutter objects from the LIVIN test set.** Top: full-scene views. Bottom: corresponding close-ups. Highlighted objects belong to the common evaluation subset used by all methods.

demand and maintains within-episode interaction history. It outputs body-frame planar velocity, yaw rate, and execution duration, alternating between action execution and new observations as needed.

OmniNav (Xue et al., 2026b) takes front, left, and right RGB views together with visual history and predicts planar waypoints and relative headings, which we convert into local GR00T commands. DualVLN (Wei et al., 2026a) takes head-camera RGB observations and visual history and follows its original System-2 planning and System-1 metric trajectory prediction pipeline. Our adapter executes each predicted local subgoal for 1 s before acquiring a new observation and replanning; the subgoal need not be reached within that interval.

E.2 Experimental Setting

All systems control the same simulated Unitree G1 humanoid on a shared set of 45 navigation tasks and environments. For each task, the start and goal locations are manually selected, and a reference path is computed with A* search on the scene NavMesh. We generate an initial route instruction from the reference path, room layout, and object semantics by selecting landmarks and route events, then converting their ordered motion and spatial relations into natural language with rule-based templates. A video following the reference path is rendered for human review, and annotators revise the instruction to ensure route–instruction consistency, reduce ambiguity among alternative paths, and make the destination clearly identifiable. Each system is evaluated three times on every task, yielding 135 episodes per system. Each episode begins with a 3 s stabilization phase, after which navigation is limited to 240 s of simulation time and 900 s of total runtime. The main evaluation metrics are introduced below.

Table 9: **Failure case analysis of navigation methods on LIVIN.** Failure categories are reported as percentages of failed episodes for each method. Failed NE denotes the mean navigation error over failed episodes.

Method	Unrecovered Blockage (%)	Fall (%)	Goal-Completion Mismatch (%)	Other Failures (%)	Failed NE (m) ↓
GPT-6 Astra (OpenAI, 2026)	1.27	17.72	77.22	3.80	**3.15**
OmniNav (Xue et al., 2026b)	69.05	2.38	15.08	13.49	7.12
DualVLN (Wei et al., 2026a)	14.73	0.00	71.32	13.95	6.22

Navigation error (NE). For execution r of task i, navigation error is

$$\mathrm{NE}_{ir} = d_{\mathrm{nav}}\left(\boldsymbol{p}_{ir}^{\mathrm{final}}, \boldsymbol{g}_i\right) m, \tag{1}$$

where $\boldsymbol{p}_{ir}^{\mathrm{final}}$ is the final planar robot position and $\boldsymbol{g}_i$ is the destination. The navigable distance d_{nav} is computed after projection onto the evaluation NavMesh using triangle-graph A* search followed by funnel refinement.

Success rate (SR). An execution is successful only if the system explicitly issues STOP and the final navigable distance to the destination is at most $\tau = 1\,\mathrm{m}$:

$$S_{ir} = \mathbb{I}\left[\textsc{Stop} \ \wedge \ \mathrm{NE}_{ir} \leq \tau\right]. \tag{2}$$

We use a relatively strict $1\mathrm{m}$ threshold because LIVIN preserves the dense object arrangements of real lived-in homes, where subsequent interaction often requires the robot to reach the immediate vicinity of a specific object or interaction region. SR is the aggregated success indicator and is reported as a percentage.

Success weighted by path length (SPL). For each execution,

$$\mathrm{SPL}_{ir} = S_{ir} \frac{L_i}{\max(L_i, P_{ir})}, \tag{3}$$

where L_i is the start-to-destination reference path length on the evaluation NavMesh and P_{ir} is the accumulated planar trajectory length of the robot base after stabilization. Aggregated SPL is reported as a percentage.

Collision counts. All systems use the same MuJoCo (Todorov et al., 2012) collision geometry at a voxel resolution of $2.5\,\mathrm{cm}$. Contacts involving multiple robot links at the same time are counted as one collision event. Contact intervals separated by less than $0.1\,\mathrm{s}$ are merged, and a new event is counted only after all eligible contacts have ended for at least $0.1\,\mathrm{s}$.

E.3 Further Analysis of Navigation Performance

Failure case analysis. To better understand the differences between navigation methods, we further analyze the observed failure patterns. We group failed episodes according to the execution evidence into unrecovered blockage, falls, goal-completion mismatch, and other failures, such as repetitive turning and execution stalls. We also report the mean navigation error (NE) over failed episodes to characterize how far the robot remains from the goal when navigation fails. The statistics are summarized in Tab. 9.

The failure statistics further support the observations in Sec. 6.1. For OmniNav, unrecovered blockage is the dominant failure mode, accounting for 69.05% of failed episodes, while DualVLN shows a much lower blockage rate but a large fraction of goal-completion mismatches. GPT-6 Astra is also dominated by goal-completion mismatch, but its failed-episode NE is substantially lower than that of DualVLN, suggesting that many failures occur near the destination and are often caused by inaccurate final stopping-region judgments than by large route deviations.

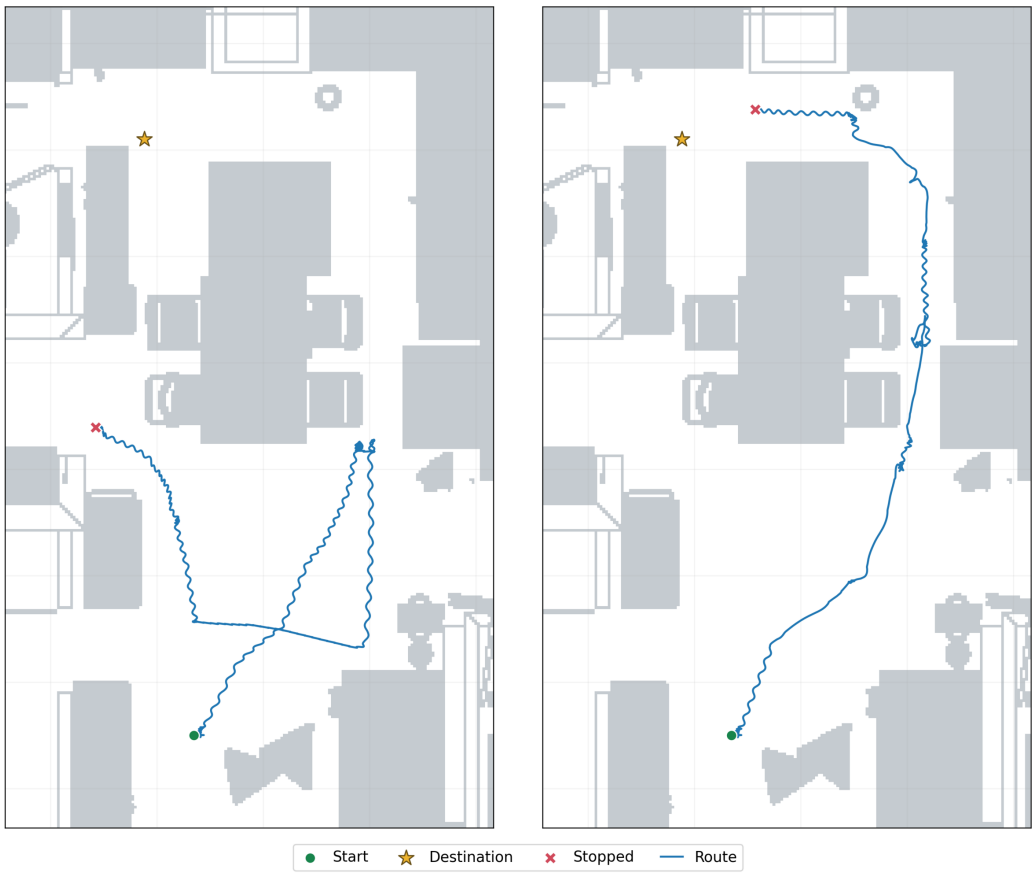


Figure 12: **Navigation trajectories with GR00T and the mixed execution setting.** Left: GPT-6 Astra+GR00T fails to pass the narrow passage on the right side of the dining room. Right: GPT-6 Astra+Mixed uses CAT for local traversal, allowing the robot to pass through the constrained right-side passage and continue toward the goal.

Analysis of the mixed setting. We further explore whether introducing an obstacle-avoidance strategy can improve navigation in constrained household spaces. To this end, we use CAT (Xue et al., 2026a), a collision-free humanoid traversal policy for cluttered environments. During navigation, GPT-6 Astra selects between GR00T and CAT based on the current context, allowing it to switch to CAT when additional obstacle-avoidance capability is needed. This mixed strategy can help in tasks that require the robot to pass through narrow or cluttered spaces, as shown in Fig. 12. However, its overall success rate is 34.81%, which is lower than that of GPT-6 Astra+GR00T. We find that, when CAT is selected for local traversal, the robot often walks sideways, with its body orientation misaligned with the direction of travel. This can make the head-mounted RGB observations seen by the high-level planner inconsistent with the robot's actual motion direction, which may affect subsequent instruction following and turning decisions.

F Loco-Manipulation Evaluation Details

F.1 Task Inventory

The benchmark comprises 35 tasks across 30 homes, grouped into object grasping and placement, object pushing and pulling, articulated-object manipulation, and long-horizon tasks with dependent subgoals. Tab. 10 lists the task instructions.

Table 10: **Loco-manipulation tasks.** Instructions are condensed from the task definitions.

Object grasping and placement (15 tasks)	
LM1	Move a saucer from the placemat to an empty spot on the table.
LM2	Grasp and lift a small green cream jar and keep holding it.
LM3	Grasp and lift a handheld massager and hold it stably.
LM4	Grasp and lift a gray insulated cup and keep holding it.
LM5	Place the wine glass on the table to the right of the placemat.
LM6	Lift a toy elephant with both hands and keep holding it.
LM7	Place a cup on the counter to the left of a chocolate jar.
LM8	Move a bottle inward, away from the table edge.
LM9	Grasp and lift a fitness ring and keep holding it.
LM10	Place an oval box on a coaster.
LM11	Move a water bottle to the counter left of the water dispenser.
LM12	Place a remote control on the coffee table to the left of a small red box.
LM13	Lift a tissue box with both hands and keep holding it.
LM14	Lift a vacuum flask off its support with both hands and hold it stably.
LM15	Grasp and lift a fitness ring from the floor and keep holding it.
Articulated-object manipulation (7 tasks)	
AM1	Pull a kitchen base-cabinet door open by its handle to allow access to the interior.
AM2	Open an oven door and then close it.
AM3	Pull out the upper drawer of an under-counter steriliser cabinet far enough to access its contents, and keep it open.
AM4	Close the open upper door of a refrigerator.
AM5	Open a countertop microwave door to allow access to the interior, and keep it open.
AM6	Close both doors at the entrance of a walk-in closet.
AM7	Pull the leftmost under-window kitchen base-cabinet door open by its handle to allow access to the interior.
Object pushing and pulling (5 tasks)	
PC1	Pull a chair backward into the designated target area.
PC2	Move a chair forward into the designated target area.
PC3	Move a study chair to clear the room exit.
PC4	Move the chair beside the dining table aside to make room for people to walk through.
PC5	Push the gray baby stroller forward.
Long-horizon tasks (8 tasks)	
LH1	Move two blocking bottles aside, then lift the blue-capped water bottle and keep holding it.
LH2	Move a pump bottle from the media console below the television to the coffee table.
LH3	Pick up a detergent bottle and place it upright on the kitchen counter beside the sink, then place a small bowl inside the sink.
LH4	Open a kitchen base-cabinet door and place the air fryer from the countertop inside.
LH5	Carry an egg from the kitchen counter to the dining table and set it down beside the tissue box.
LH6	Open the closet door, enter the walk-in closet, and touch the black cloth with one hand.
LH7	Carry a cushion from the living-room sofa to the bed in a bedroom.
LH8	Carry a metal thermos from the bedside table to the living-room coffee table and set it down upright.

F.2 Simulation and Embodiment

All tasks use a simulated Unitree G1 humanoid with 29 body joints (12 leg, 3 waist, and 14 arm joints) and two 7-DoF Dex3 hands. Following SIMPLE (Wei et al., 2026c), we use MuJoCo for physics simulation and Isaac Sim for rendering, with a physics time step of 5 ms. The simulation pauses during model inference and resumes for action execution.

F.3 Evaluated Methods

DeepSeek-V4.1-Flash (DeepSeek-AI, 2026) and GPT-6 Astra (OpenAI, 2026) share a turn-based tool-calling interface with identical system instructions and controller. Both models receive the task instruction, robot state information, and head-camera RGB images, and issue tool calls to observe the environment, execute whole-body motions, wait without issuing commands, or finish the episode. Each model waits for the current action to complete before issuing the next call. Both models are accessed through their respective official APIs with medium reasoning effort.

Ψ_0. We evaluate Ψ_0 (Wei et al., 2026b) using its released checkpoints and native controllers, without fine-tuning on LIVIN.

F.4 Scene and Object Modelling

Each scene uses the reconstructed meshes described in Sec. 4, exported from Blender for simulation. Collision geometry is computed with CoACD (Wei et al., 2022), which adaptively decomposes concave meshes into convex hulls. Physical properties (mass, centre of mass, inertia tensor, and contact friction) are initially proposed by GPT-6 Astra, and subsequently reviewed and corrected by the authors.

F.5 Evaluation Metrics

Success rate (SR) is the percentage of trials meeting the task-specific success criteria within each task family. For each long-horizon trial, process score (PS) is the percentage of equally weighted physical subgoals completed with their prerequisites satisfied. We report mean PS across trials on a 0–100 scale.