AIrchitect: Domain-Grounded Interactive 3D World Generation for Social Robotics

Volodymyr Shcherbyna[1,2], Duc Anh Do[2], Duc Tai Nguyen[2], and Linh Kästner[2]

[1]Technische Universität Berlin (TUB), Germany
[2]Singapore Management University (SMU), Singapore

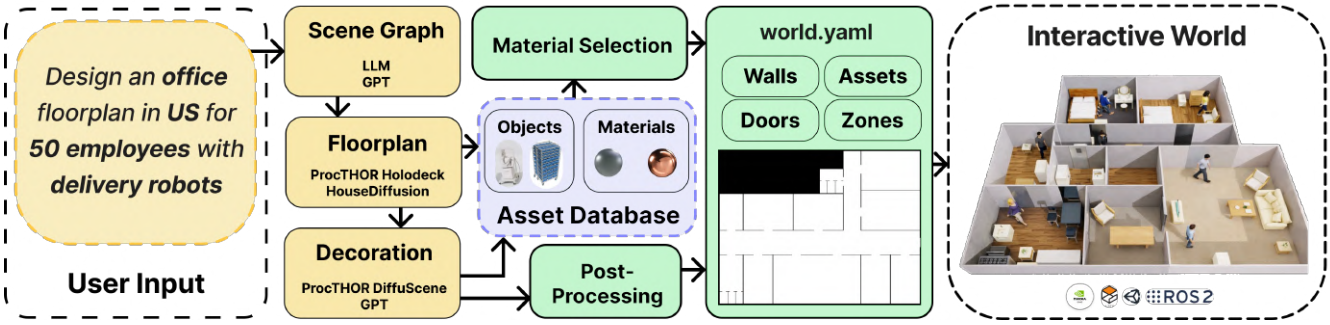

Fig. 1. *AIrchitect Overview*: End-to-end pipeline from natural language prompt or floorplan image to 3D environment. The platform combines generation backends with an interactive web frontend (Fig. 2) and direct export to Gazebo and Isaac Sim.

***Abstract*— Training and validating embodied AI requires realistic simulated indoor environments, yet existing generation methods are difficult to combine and rarely support converting real floorplans into simulation. We introduce AIrchitect, an open-source platform that generates interactive 3D indoor environments from natural language prompts or floorplan images. Its microservice architecture lets users swap between diffusion-based, procedural, and LLM-based backends for both floorplan generation and asset placement, all behind a common API. A web-based frontend exposes the full pipeline from domain selection, through interactive editing, to export. We generate 450 worlds across residential, hospital, and office domains and report structural and perceptual quality metrics. The repository is publicly available at github.com/voshch/AIrchitect.**

I. Introduction

Generating realistic digital twins of indoor environments is increasingly central to robotics research: simulated worlds are where embodied AI agents are trained and validated, and their fidelity directly affects transfer to real hardware [1].

Several families of generation methods now exist: procedural generators [2], diffusion models [3], [4], and LLM-based planners [5], but each lives in its own codebase with its own data formats, making it hard to mix or compare them. Most also work only in the Sim2Real direction, generating worlds from scratch; few support the reverse path of converting an existing floorplan into a simulable environment.

We introduce *AIrchitect* (Fig. 1), an open-source platform that addresses both problems:

1) *Modularity.* A microservice architecture exposes each pipeline stage behind a uniform REST API: scene-graph generation, floorplan layout, asset placement. Backends such as HouseDiffusion [3], ProcTHOR [2], and Holodeck [5] are interchangeable, so researchers can mix methods or benchmark them against each other.
2) *Dual input.* The platform accepts either a text prompt or a floorplan image, covering both the Sim2Real and Real2Sim directions within a single pipeline.
3) *Accessibility.* A web frontend guides users through a six-stage workflow with interactive editing and preview at every step.

AIrchitect currently supports residential, hospital, and office domains and exports directly to Gazebo and Isaac Sim via ROS 2 [6], [7]. We evaluate the platform on 450 generated worlds and report both structural and perceptual metrics.

II. Related Work

Generating indoor environments for embodied AI involves trade-offs between realism, physical soundness, and scalability. We briefly survey the main threads; Table I shows which methods AIrchitect integrates.

a) Scan and designer-created datasets: Real-world scans such as HM3D [8] and Matterport3D [9] preserve spatial complexity but yield non-watertight meshes that need heavy post-processing. Designer datasets like 3D-FRONT [10] provide clean CAD assets yet cover only residential interiors. Simulation platforms (Habitat [11], AI2-THOR [12]) wrap these datasets with physics engines but inherit the same domain limits.

b) Procedural and diffusion-based generation: ProcTHOR [2] and Infinigen [13] scale through procedural rules. HouseDiffusion [3] generates high-fidelity residential floorplans via graph-transformer diffusion. DiffuScene [4] produces dense room layouts through denoising, and RoomDreamer [14] synthesizes coherent indoor textures.

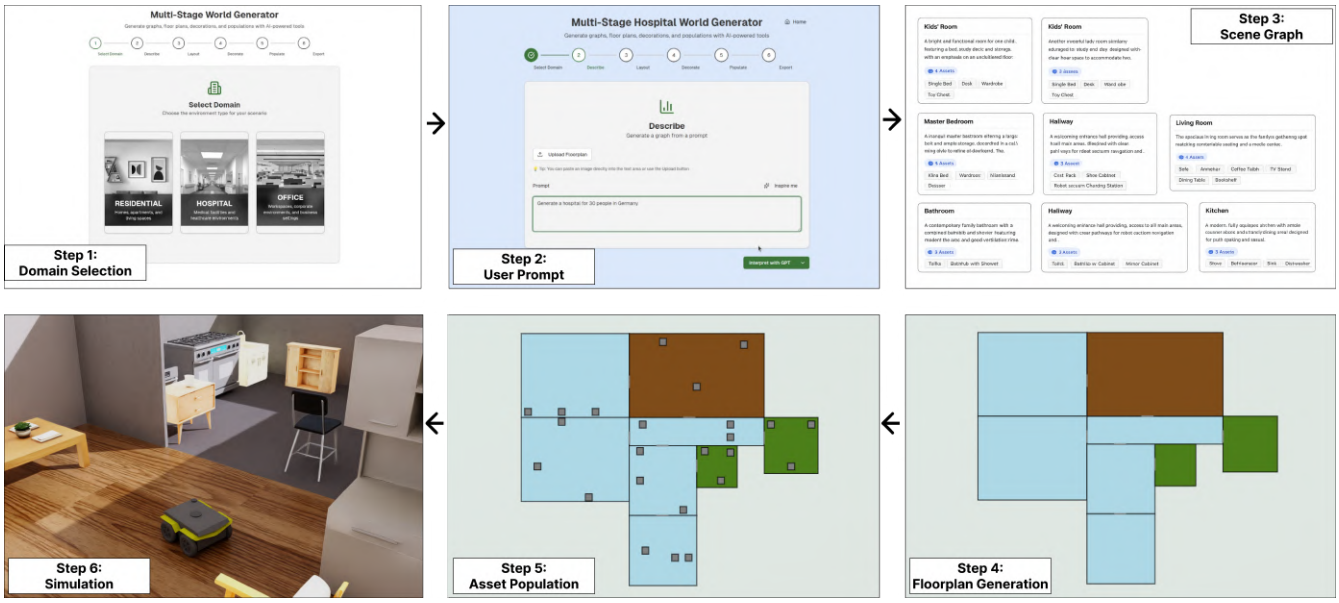

Fig. 2. *AIrchitect Web Frontend*: The six-stage wizard guides users from domain selection through export. At each stage, the user can inspect results, switch backends via the split button, regenerate, or navigate back. The scene graph stage supports full interactive editing of rooms, connections, and assets.

AIrchitect integrates HouseDiffusion, ProcTHOR, and DiffuScene as interchangeable backends so users can pick the best method for each stage.

c) Agentic and LLM-based generation: LayoutGPT [15] treats layout synthesis as an LLM reasoning task. PhyScene [16] adds differentiable physical constraints, and SAGE [17] uses a physics critic for iterative correction. Holodeck [5] combines LLM planning with asset retrieval for multi-room generation and is also available inside AIrchitect.

d) Scene-graph representations: 3D Scene Graphs [18] and SayPlan [19] use hierarchical spatial representations for robotic reasoning and task planning, but assume the 3D world already exists. AIrchitect, by contrast, *generates* both the scene graph and the underlying environment.

III. System Design

AIrchitect transforms a text prompt or a floorplan image into a fully interactive 3D simulation environment. Each pipeline stage runs as an independent Docker service behind a REST API, so individual components can be developed, replaced, or scaled independently.

A. User Workflow

AIrchitect provides a Next.js-based web application that guides users through the full generation workflow (Fig. 2). The interface is organized as a six-stage wizard with a progress indicator, where each stage presents the current output for inspection alongside controls for proceeding, regenerating, or navigating back.

a) Domain selection and input: The workflow begins with domain selection, where the user clicks one of three domain cards (residential, hospital, office), immediately creating a new session. The *describe* stage supports two input modalities:

- *Text prompt.* The user types a natural language description of the desired environment (e.g., *"Generate a hospital for 30 people in Germany"*). An "Inspire me" button can also generate a domain-appropriate prompt automatically.
- *Floorplan image.* The user uploads or pastes a photograph or scan of an existing floorplan. A vision model (Gemini) extracts room layouts, identifies room types and object annotations, and produces the same scene graph, so all later stages are input-agnostic.

b) Scene graph editing: The web frontend renders the resulting scene graph as an interactive React Flow canvas. Each room appears as a card showing the room type, a short textual description, and editable asset tags. Users can create new rooms, delete existing ones, draw adjacency connections, edit each room's asset list, and freely rearrange the spatial layout by dragging nodes. This editing step is particularly valuable for steerable generation, where the user can modify the automatically extracted graph before proceeding.

c) Floorplan, decoration, and export: After submitting the graph, the *decorate* stage renders the generated floorplan as a 2D canvas with color-coded room polygons. The *populate* stage reuses the same canvas, now showing placed assets as icons within their rooms, with an additional prompt field for describing pedestrian scenarios. The *export* stage displays the final world with both assets and pedestrian waypoints (shown as colored dots), offering ZIP download of the `world.yaml`. At each generation step, a split button lets users choose which backend to invoke, and a "Regenerate" button re-runs the current stage.

B. Generation Pipeline

Table I lists all currently available backends with their characteristics. The generation pipeline proceeds through the

following stages.

a) Text-to-graph and image-to-graph: A natural language prompt or floorplan image is transformed into a hierarchical scene graph $G = (V, E)$. The graph has two levels: an upper level where vertices V represent rooms and edges E encode adjacency through doorways, and a lower level where each room node contains a sub-graph of assets defined by textual descriptions, expected sizes, and colors. For text input, the LLM module uses domain-specific few-shot examples to produce the graph, and an asset enrichment pass fills sparse rooms using room-type keyword hints.

For image input (Fig. 4), the pipeline first runs two asynchronous vision model calls to detect the pixel-to-meter scale and identify the indoor area with a room count estimate. The image is then masked, cropped to the indoor region, and split into tiles to handle large or high-resolution floorplans. Per-tile extraction identifies room bounding boxes with asset annotations and door bounding boxes in parallel. A post-processing stage converts pixel coordinates to meters using the detected scale, merges room segments across tile boundaries, computes corridor areas from the remaining space, and assembles the final scene graph.

b) Floorplan generation: The scene graph is passed to one of four interchangeable floorplan backends (Table I). The default RAG Floor backend uses an LLM to generate room and door polygon coordinates from the graph. A post-processing step constructs Shapely [21] geometries, derives wall segments from boundary differences between room polygons and door openings, and assembles the final world description. Alternatively, users can select HouseDiffusion for its high geometric fidelity on residential layouts (trained on CubiCasa5K [20], a dataset of $\sim$5000 real-world floor plan scans), ProcTHOR for procedural scalability, or Holodeck for LLM-guided multi-domain generation.

c) Asset decoration: Each room is populated with 3D models retrieved from the asset database (Section III-C). Given a room polygon, door geometry, and the room's asset sub-graph, the decoration module outputs collision-free poses for all assets. The default LLM-guided backend receives candidate asset metadata (bounding boxes, face directions, descriptions) and proposes positions validated against hard constraints: no mutual overlap, full containment within the room boundary, and minimum clearance around doors. ProcTHOR and DiffuScene [4] are available as alternatives.

d) Post-processing and export: A post-processing pipeline converts the 2D layout into a 3D environment. Room polygons are projected onto an occupancy grid with doorways carved as free space. Square probe regions beyond each doorway detect and seal exterior openings. The resulting occupancy map is extruded into a 3D wall model via wavefront expansion. All placed assets and their HOI/ROI annotations are gathered into a central `world.yaml`. The pipeline also outputs ground-truth artifacts: room and doorway segmentations with semantic labels, a scene graph compatible with planners such as SayPlan [19], and entry/exit points for spawning dynamic entities in crowd simulations. The final world is packaged as a ROS 2-compatible zip that loads into Gazebo and Isaac Sim [7] without manual conversion.

C. Asset Database

The asset database is an automatically processed dataset of 3D models (multi-format, primarily USDZ) queryable via free-text search. Each domain maintains a ChromaDB vector store built from model annotations. Text embeddings are computed using spaCy with a pre-trained GloVe model; records are retrieved via approximate nearest-neighbor search with distance-weighted random selection from the top matches.

For each asset, the database computes and stores the following annotations:

- *Bounding box:* axis-aligned 3D extents in meters;
- *Material list:* physical surface materials (e.g., leather, steel, wood);
- *Color palette:* dominant colors (e.g., white, silver);
- *Human/Robot–object interactions (HOI/ROI):* affordance tags (e.g., *lie*, *sit*, *operate*);
- *Face direction:* canonical front-facing axis ($+x$, $-x$, $+y$, $-y$) for consistent placement orientation;
- *Semantic tags:* domain and category labels (e.g., *office*, *hospital*, *furniture*);
- *Free-text description:* a short natural-language description (e.g., "Gurney", "Armchair").

All annotations are concatenated into a single text embedding for nearest-neighbor retrieval. During decoration, the matched metadata (e.g, bounding box dimensions and face direction) feeds directly into the placement backend for pose computation.

The current database spans three domains:

Domain	Office	Hospital	Residential	Total
# 3D Assets	139	71	49	259

Adding a new domain requires only placing annotated 3D models in the expected directory structure and running the build pipeline, which generates the vector store and model conversions automatically.

D. Architecture

All services are containerized with Docker and orchestrated via Docker Compose. The shared API layer supports both synchronous and deferred (callback-based) request patterns. A common Python package defines the type system: scene graphs, world descriptions, obstacle poses. This ensures that any backend combination interoperates without glue code.

IV. Evaluation

A. Generation Statistics

To characterize the platform at scale, we generated 450 worlds using the default pipeline (RAG Floor + GPT decoration), with 50 samples each at requested room counts of 5, 8, and 10 for all 3 domains (total 450). Metrics are extracted automatically from the `world.yaml` files; areas are computed with the Shoelace formula.

Fig. 3. *Generated Environments*: Example outputs across three domains rendered in Isaac Sim. Left to right: hospital, office, residential.

TABLE I

AVAILABLE BACKENDS PER PIPELINE STAGE.

Stage	Backend	Type	Domains	Speed	Description
Scene Graph	*GPT*	LLM (Gemini/OpenAI)	All	fast	Few-shot prompted; domain-specific examples; supports both text and image input
	Local LLM	Fine-tuned T5	Residential	fast	offline trained on hand-annotated scene graphs
Floorplan	HouseDiffusion	GNN diffusion	Residential	med.	Graph-transformer w/ discrete/continuous denoising [3]; trained on CubiCasa5K [20]
	ProcTHOR	Procedural	Res./Hosp.	fast	Rule-based procedural generation [2]
	Holodeck	LLM-guided	All	slow	LLM planning with Objaverse retrieval [5]
	RAG Floor	LLM (Gemini)	All	med.	Polygon generation w/ optional constraint injection
decoration	*GPT*	LLM-guided	All	med.	Semantic retrieval + LLM-proposed collision-free placement
	ProcTHOR	Procedural	Res./Hosp.	fast	Rule-based placement [2]
	DiffuScene	Diffusion	Residential	slow	Denoising diffusion for room layouts [4]

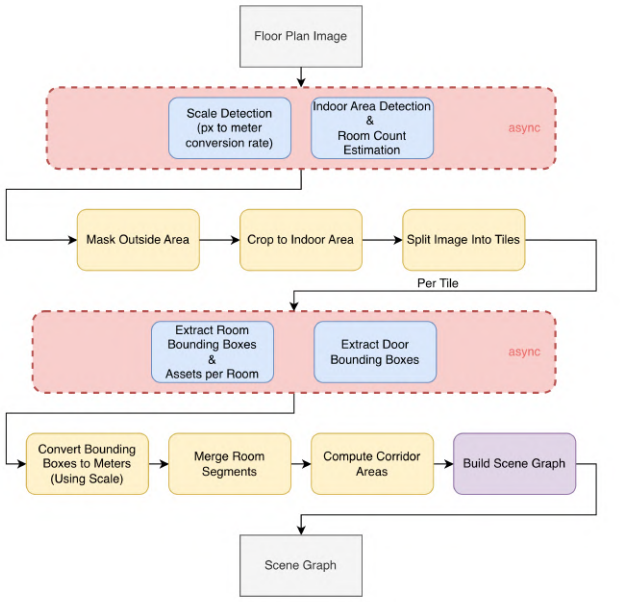

Fig. 4. *Image-to-Graph Pipeline*: The Real2Sim path for extracting a scene graph from a floorplan image. Asynchronous vision model calls detect the pixel-to-meter scale and indoor area in parallel. The image is masked, cropped, and tiled; per-tile extraction identifies room and door bounding boxes. Post-processing merges segments, converts coordinates to meters, computes corridor areas, and assembles the final scene graph.

Table II summarizes the results. Room-count fidelity is perfect: realized counts always match the request. Office

TABLE II

GENERATION STATISTICS ACROSS 450 WORLDS (n=50 PER CONFIGURATION) USING THE DEFAULT PIPELINE.

Configuration	Objs	Doors	Walls	Area (m^2)	Objs/Rm
Residential (5-rm)	30	5	30	95	6.1
Residential (8-rm)	48	9	50	158	6.0
Residential (10-rm)	60	11	62	222	6.0
Office (5-rm)	45	6	32	287	8.9
Office (8-rm)	64	9	49	347	7.9
Office (10-rm)	72	10	60	365	7.2
Hospital (5-rm)	35	5	30	114	7.0
Hospital (8-rm)	56	8	48	231	7.0
Hospital (10-rm)	69	10	61	246	6.9

and hospital worlds are substantially larger than residential ones, reflecting their different spatial programs. Object density peaks in offices (up to 8.9 per room), consistent with workstation-heavy layouts, while residential scenes are sparser. Hospitals show the highest topological complexity with more doors and wall segments, reflecting corridor-based circulation.

B. Perceptual Quality

We evaluate 18 residential worlds using automated perceptual metrics from the SceneEval protocol [22]. A vision-language model scores each world on four axes: *Layout* (spatial plausibility), *Visual* (rendering and material coherence), *VQA Accuracy* (object identification correctness), and *Scene*

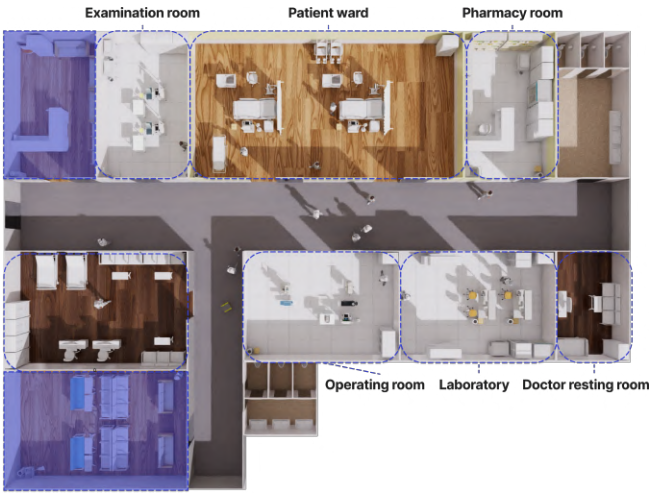

Fig. 5. *Simulator Deployment*: A generated hospital environment loaded into Isaac Sim. Key rooms highlighted and annotated.

Rating (holistic quality). We also report Layout-FID as a distributional distance to real floorplans.

Table III reports the results. The default AIrchitect pipeline (RAG Floor + GPT decoration) scores 8.33/10 on layout and 7.61/10 on visual quality, with the highest scene rating (8.63) among all configurations. VQA accuracy (0.400) indicates that placed objects are identifiable, though finer placement could be improved. Holodeck scores well on layout (8.46) but was evaluated without a population stage, so no scene rating is available. DiffuScene achieves the best log L-FID (3.40 vs. 3.73), expected given its training on real floorplan distributions; AIrchitect's slightly higher value reflects LLM-generated rather than learned layouts. ProcTHOR's high log L-FID (7.93) and low density (0.52 objects/room) follow from its procedural strategy, which favors speed over distributional fidelity. HouseDiffusion produces geometrically detailed plans but scores lowest on layout and visual metrics, likely because its small output areas ($40.5\,\mathrm{m}^2$) limit the variety of configurations evaluated.

As new backends are added, the same protocol allows direct comparison without extra setup.

C. Qualitative Observations

Fig. 3 shows representative outputs. Layouts preserve sensible adjacency: private rooms are separated from public circulation, service areas sit near the spaces they serve, and entries connect naturally to corridors. Object placement matches room semantics: workstation clusters in offices, treatment equipment near patient beds, seating oriented toward entertainment areas in living rooms.

Because backends are interchangeable, users can combine them freely, for example generating a floorplan with HouseDiffusion for geometric fidelity and populating it with the LLM-guided backend for context-aware asset selection.

All worlds export to both target simulators (Fig. 5), with semantic zone labels in the `world.yaml` ready for navigation, task planning, and benchmarking.

V. Conclusion

We presented AIrchitect, a platform that generates interactive 3D indoor environments for robotics from text prompts or floorplan images. Its microservice design lets researchers swap and combine generation approaches within a single pipeline, while a web frontend makes the workflow accessible to non-experts.

Across 450 generated worlds in three domains, the default pipeline achieves strong layout and visual quality scores and exports directly to Gazebo and Isaac Sim. A complete environment can be produced in under two minutes from prompt to physics-enabled simulation.

a) Limitations: The main limitation is the dependence on a pre-existing 3D asset database, which may not cover every target domain. Wall and floor texturing still requires manual adjustment. Backends that do not natively consume scene graphs require an adapter layer, which may limit their performance.

b) Future work: Planned extensions include procedural asset generation to reduce reliance on curated model libraries, conversational refinement of generated worlds, and tighter coupling with task-planning and navigation stacks. We are also exploring constrained-diffusion floorplan backends and support for multi-floor and outdoor environments.

c) Community and collaboration: Adding a new backend requires implementing a single REST endpoint against the shared type system; adding a domain requires annotated 3D models and a few-shot prompt set. Because evaluation metrics are extracted automatically from the generated `world.yaml` files, any new method can be benchmarked against existing ones with no extra tooling. Table III already demonstrates this for four backends. We encourage the community to contribute backends, assets, and domains.

References

[1] S. Höfer, K. Bekris, A. Handa, *et al.*, "Sim2real in robotics and automation: Applications and challenges," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 2, pp. 398–400, 2021.

[2] M. Deitke, E. VanderBilt, A. Herrasti, *et al.*, "ProcTHOR: Large-scale embodied AI using procedural generation," 2022.

[3] M. A. Shabani, S. Hosseini, and Y. Furukawa, "HouseDiffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising," 2022.

[4] J. Tang, Y. Nie, L. Markhasin, A. Dai, J. Thies, and M. Nießner, "DiffuScene: Denoising diffusion models for generative indoor scene synthesis," 2024.

[5] Y. Yang, F.-Y. Sun, L. Weihs, *et al.*, "Holodeck: Language guided generation of 3d embodied AI environments," 2024.

[6] L. Kästner, T. Buiyan, L. Jiao, T. A. Le, X. Zhao, Z. Shen, and J. Lambrecht, "Arena-rosnav: Towards deployment of deep-reinforcement-learning-based obstacle avoidance into conventional autonomous navigation systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6456–6463.

[7] L. Kästner, V. Shcherbyna, H. Soh, *et al.*, "Demonstrating Arena 5.0: A Photorealistic ROS2 Simulation Framework for Developing and Benchmarking Social Navigation," in *Robotics: Science and Systems*, Los Angeles, CA, USA, 2025.

[8] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, *et al.*, "Habitat-matterport 3d dataset (HM3D): 1000 large-scale 3d environments for embodied AI," 2021.

[9] A. Chang, A. Dai, T. Funkhouser, *et al.*, "Matterport3d: Learning from RGB-D data in indoor environments," 2017.

[10] H. Fu, B. Cai, L. Gao, *et al.*, "3d-front: 3d furnished rooms with layouts and furniture," 2021.

TABLE III

PERCEPTUAL QUALITY METRICS FOR 18 RESIDENTIAL WORLDS GENERATED WITH THE DEFAULT PIPELINE, EVALUATED VIA AUTOMATED VLM ASSESSMENT. FP=FLOORPLAN, D=DECORATION.

Method	n	Quantitative Metrics						Perceptual Metrics			
		Objects	Obj./Room	Doors	Area (m^2)	Obj. Spacing (m)	log Layout-FID	Layout	Visual	VQA Acc.	Scene Rating (P)
HouseDiffusion (FP)	18	-	-	8.31	40.5	-	4.72	4.48	4.60	0.313	-
Holodeck (FP)	18	-	-	7.00	154.4	-	4.11	**8.46**	7.17	0.353	-
ProcTHOR (FP+D)	18	3.73	0.52	8.11	93.87	6.41	7.93	5.50	5.78	0.333	8.09
DiffuScene (D)	18	46.60	6.33	-	-	8.58	**3.40**	7.78	7.22	0.389	8.33
AIrchitect defaults (FP+D)	18	54.56	7.06	7.67	244.3	9.19	3.73	8.33	**7.61**	**0.400**	**8.63**

[11] M. Savva, A. Kadian, O. Maksymets, *et al.*, "Habitat: A platform for embodied AI research," 2019.

[12] E. Kolve, R. Mottaghi, W. Han, *et al.*, "AI2-THOR: An interactive 3d environment for visual AI," 2022.

[13] A. Raistrick, L. Lipson, Z. Ma, *et al.*, "Infinite photorealistic worlds using procedural generation," 2023.

[14] L. Song, L. Cao, H. Xu, *et al.*, "RoomDreamer: Text-driven 3d indoor scene synthesis with coherent geometry and texture," 2023.

[15] W. Feng, W. Zhu, T.-J. Fu, *et al.*, "LayoutGPT: Compositional visual planning and generation with large language models," 2023.

[16] Y. Yang, B. Jia, P. Zhi, and S. Huang, "PhyScene: Physically interactable 3d scene synthesis for embodied AI," 2024.

[17] H. Xia, X. Li, Z. Li, *et al.*, "SAGE: Scalable agentic 3d scene generation for embodied AI," 2026.

[18] I. Armeni, Z.-Y. He, J. Gwak, *et al.*, "3d scene graph: A structure for unified semantics, 3d space, and camera," 2019.

[19] K. Rana, J. Haviland, S. Garg, *et al.*, "SayPlan: Grounding large language models using 3d scene graphs for scalable robot task planning," 2023.

[20] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, and J. Kannala, "CubiCasa5K: A dataset and an improved multi-task model for floorplan image analysis," 2019.

[21] S. Gillies, C. van der Wel, J. Van den Bossche, *et al.*, "Shapely (2.1.2)," 2025. [Online]. Available: https://doi.org/10.5281/zenodo.17193310

[22] H. I. I. Tam, H. I. D. Pun, A. X. T. Wang, A. X. Chang, and M. Savva, "SceneEval: Evaluating semantic coherence in text-conditioned 3d indoor scene synthesis," 2025.