HOLODECK: Language Guided Generation of 3D Embodied AI Environments

Yue Yang*[1], Fan-Yun Sun*[2], Luca Weihs*[4], Eli Vanderbilt[4], Alvaro Herrasti[4],
Winson Han[4], Jiajun Wu[2], Nick Haber[2], Ranjay Krishna[3,4], Lingjie Liu[1],
Chris Callison-Burch[1], Mark Yatskar[1], Aniruddha Kembhavi[3,4], Christopher Clark[4]
[1]University of Pennsylvania, [2]Stanford University,
[3]University of Washington, [4]Allen Institute for Artificial Intelligence
yueyang1996.github.io/holodeck/

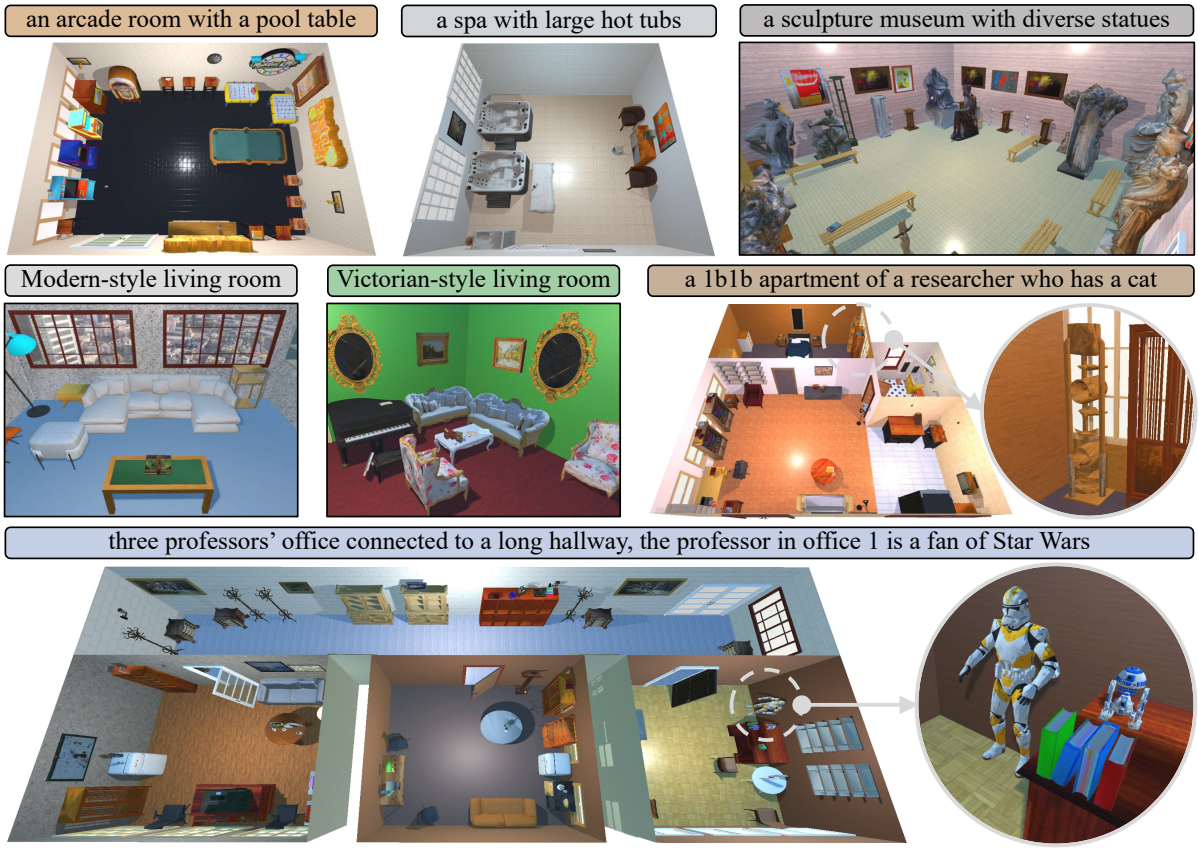

Figure 1. Example outputs of HOLODECK—a large language model powered system, which can generate diverse types of environments (arcade, spa, museum), customize for styles (Victorian-style), and understand fine-grained requirements ("has a cat", "fan of Star Wars").

Abstract

3D simulated environments play a critical role in Embodied AI, but their creation requires expertise and extensive manual effort, restricting their diversity and scope. To mitigate this limitation, we present HOLODECK*, a system that generates 3D environments to match a user-supplied prompt fully automatedly.* HOLODECK *can generate diverse scenes, e.g., arcades, spas, and museums, adjust the designs for styles, and can capture the semantics of complex queries such as "apartment for a researcher with a cat" and "office of a professor who is a fan of Star Wars".* HOLODECK *leverages a large language model (i.e., GPT-4) for common sense knowledge about what the scene might look like and uses a large collection of 3D assets from Objaverse to populate the scene with diverse objects. To address the challenge of positioning objects correctly, we prompt GPT-4 to generate spatial relational constraints between objects and then optimize the layout to satisfy those constraints. Our large-scale*

*Equal technical contribution. Work done while at PRIOR@AI2.

human evaluation shows that annotators prefer HOLODECK *over manually designed procedural baselines in residential scenes and that* HOLODECK *can produce high-quality outputs for diverse scene types. We also demonstrate an exciting application of* HOLODECK *in Embodied AI, training agents to navigate in novel scenes like music rooms and daycares without human-constructed data, which is a significant step forward in developing general-purpose embodied agents.*

1. Introduction

The predominant approach in training embodied agents involves learning in simulators [7, 20, 23, 35, 40, 51]. Generating realistic, diverse, and interactive 3D environments plays a crucial role in the success of this process.

Existing Embodied AI environments are typically crafted through manual design [5, 12, 23, 24], 3D scanning [7, 38, 40], or procedurally generated with hard-coded rules [6]. However, these methods require considerable human effort that involves designing a complex layout, using assets supported by an interactive simulator, and placing them into scenes while ensuring semantic consistency between the different scene elements. Therefore, prior work on producing 3D environments mainly focuses on limited environment types. To move beyond these limitations, recent works adapt 2D foundational models to generate 3D scenes from text [10, 16, 53]. However, these models often produce scenes with significant artifacts, such as mesh distortions, and lack the interactivity necessary for Embodied AI. Moreover, there are models tailored for specific tasks like floor plan generation [17, 42] or object arrangement [33, 49]. Although effective in their respective domains, they lack overall scene consistency and rely heavily on task-specific datasets.

In light of these challenges, we present **HOLODECK**, a language-guided system built upon AI2-THOR [23], to automatically generate diverse, customized, and interactive 3D embodied environments from textual descriptions. Shown in Figure 2, given a description (e.g., *a 1b1b apartment of a researcher who has a cat*), HOLODECK uses a Large Language Model (GPT-4 [32]) to design the floor plan, assign suitable materials, install the doorways and windows and arrange 3D assets coherently in the scene using constraint-based optimization. HOLODECK chooses from over 50K diverse and high-quality 3D assets from Objaverse [8] to satisfy a myriad of environment descriptions.

Motivated by the emergent abilities of Large Language Models (LLMs) [48], HOLODECK exploits the commonsense priors and spatial knowledge inherently present in LLMs. This is exemplified in Figure 1, where HOLODECK creates diverse scene types such as *arcade*, *spa* and *museum*, interprets specific and abstract prompts by placing relevant objects appropriately into the scene, e.g., an "R2-D2"[1] on the desk for "a fan of Star Wars". Beyond object selection and layout design, HOLODECK showcases its versatility in style customization, such as creating a scene in a "Victorian-style" by applying appropriate textures and designs to the scene and its objects. Moreover, HOLODECK demonstrates its proficiency in spatial reasoning, like devising floor plans for "three professors' offices connected by a long hallway" and having regular arrangements of objects in the scenes. Overall, HOLODECK offers a broad coverage approach to 3D environment generation, where textual prompts unlock new levels of control and flexibility in scene creation.

The effectiveness of HOLODECK is assessed through its scene generation quality and applicability to Embodied AI. Through large-scale user studies involving 680 participants, we demonstrate that HOLODECK significantly surpasses existing procedural baseline PROCTHOR [6] in generating residential scenes and achieves high-quality outputs for various scene types. For the Embodied AI experiments, we focus on HOLODECK's application in aiding zero-shot object navigation in previously unseen scene types. We show that agents trained on scenes generated by HOLODECK can navigate better in novel environments (e.g., *Daycare* and *Gym*) designed by experts.

To summarize, our contributions are three-fold: (1) We propose HOLODECK, a language-guided system capable of generating diverse, customized, and interactive 3D environments based on textual descriptions; (2) The human evaluation validates HOLODECK's capability of generating residential and diverse scenes with accurate asset selection and realistic layout design; (3) Our experiments demonstrate that HOLODECK can aid Embodied AI agents in adapting to new scene types and objects during object navigation tasks.

2. Related Work

Embodied AI Environments. Previous work mainly relies on 3D artists to design the environments [5, 12, 22–24, 35, 51], which is hard to scale up or construct scenes from 3D scans [38, 40, 43] to reduce human labor, but scenes are less interactive. The procedural generation framework PROCTHOR [6] showcases its potential to generate large-scale interactive environments for training embodied agents. Phone2Proc [7] uses a phone scan to create training scenes that are semantically similar to the desired real-world scene. A concurrent work, RoboGen [47], proposes to train robots by generating diversified tasks and scenes. These works parallel our concept, HOLODECK, which aims to train generalizable embodied agents and presents an avenue for further exploration in text-driven 3D interactive scene generation.

Large Language Model for Scene Design. Many works on scene design either learn spatial knowledge priors from existing 3D scene databases [3, 27, 44–46, 49, 54] or leverage user input and refine the 3D scene iteratively [2, 4]. However, having to learn from datasets of limited categories such

[1] A fictional robot character in the Star Wars.

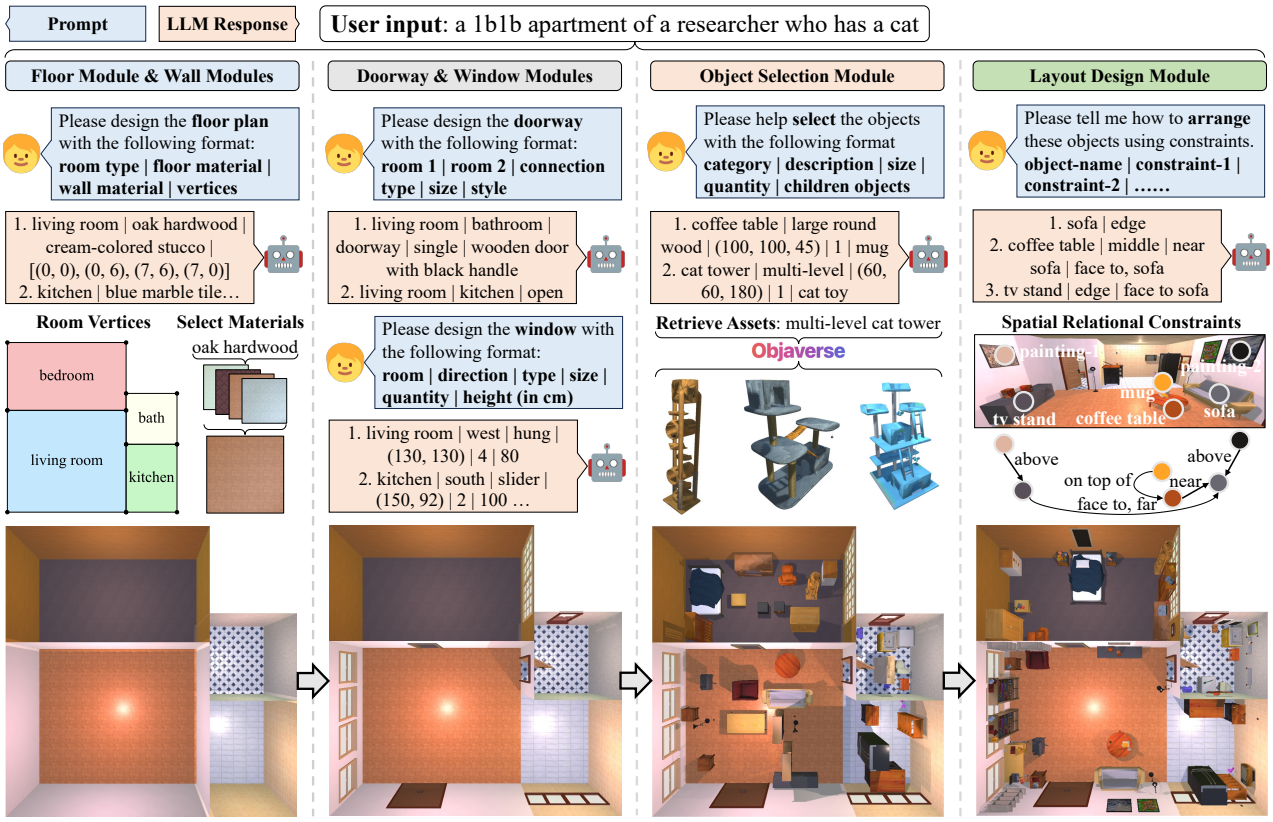

Figure 2. Given a text input, HOLODECK generates the 3D environment through multiple rounds of conversation with an LLM.

as 3D-FRONT [11] restricts their applicability. Recently, Large Language Models (LLMs) were shown to be useful in generating 3D scene layouts [9, 26]. However, their methods of having LLMs directly output numerical values can yield layouts that defy physical plausibility (e.g., overlapping assets). In contrast, HOLODECK uses LLMs to sample spatial relational constraints and a solver to optimize the layout, ensuring physically plausible scene arrangements. Our human study shows a preference for HOLODECK-generated layouts over those generated end-to-end by LLMs. (see Sec 4.3).

Text-driven 3D Generation. Early endeavors in 3D generation focus on learning the distribution of 3D shapes and/or textures from category-specific datasets [14, 30, 50, 52, 55]. Subsequently, the advent of large vision-language models like CLIP [37] enables zero-shot generation of 3D textures and objects [13, 18, 25, 28, 29, 34]. These works excel at generating 3D objects but struggle to generate complex 3D scenes. More recently, emerging works generate 3D scenes by combining pre-trained text-to-image models with depth prediction algorithms to produce either textured meshes or NeRFs [10, 16, 53]. However, these approaches yield 3D representations that lack modular composability and interactive affordances, limiting their use in embodied AI. In contrast, HOLODECK utilizes a comprehensive 3D asset database to generate semantically precise, spatially efficient, and interactive 3D environments suitable for training embodied agents.

3. HOLODECK

HOLODECK is a promptable system based on AI2-THOR [6, 23], enriched with massive assets from Objaverse [8], which can produce diverse, customized, and interactive Embodied AI environments with the guidance of large language models.

As shown in Figure 2, HOLODECK employs a systematic approach to scene construction, utilizing a series of specialized modules: (1) the *Floor & Wall Module* develop floor plans, constructs wall structures and selects appropriate materials for the floors and walls; (2) the *Doorway & Window Module* integrates doorways and windows into the environment; (3) the *Object Selection Module* retrieves appropriate 3D assets from Objaverse, and (4) the *Constraint-based Layout Design Module* arranges the assets within the scene by utilizing spatial relational constraints to ensure that the layout of objects is realistic.

In the following sections, we introduce our prompting approach that converts high-level user natural language specifications into a series of language model queries for constructing layouts. We then provide a detailed overview of each module shown in Figure 2 and how they contribute to the final scene. Finally, we illustrate how HOLODECK leverages Objaverse assets to ensure diversity in scene creation and efficiency for Embodied AI applications. Comprehensive details of HOLODECK can be found in the supplement.

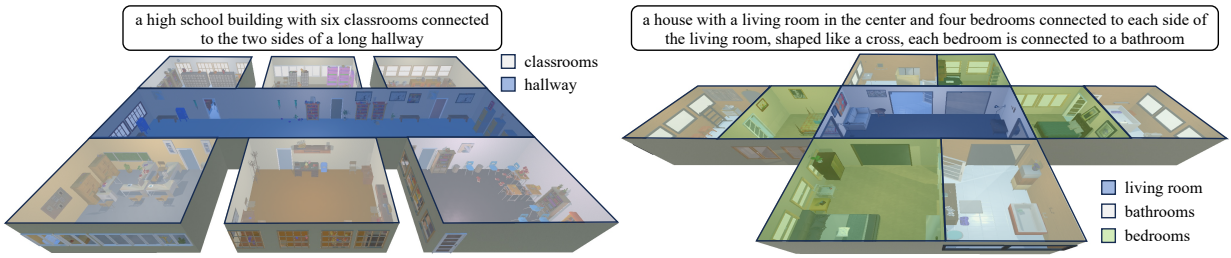

Figure 3. **Floorplan Customizability.** HOLODECK can interpret complicated input and craft reasonable floor plans correspondingly.

Figure 4. **Material Customizability.** HOLODECK can select appropriate floor and wall materials to make the scenes more realistic.

Figure 5. **Door & window Customizability.** HOLODECK can adjust the size, quantity, position, etc., of doors & windows based on the input.

Overall Prompt Design. Each module in Figure 2 takes information from a language model and converts it to elements included in the final layout. An LLM prompt is designed for each module with three elements: (1) *Task Description*: outlines the context and goals of the task; (2) *Output Format*: specifies the expected structure and type of outputs and (3) *One-shot Example*: a concrete example to assist the LLM's comprehension of the task. The text within the blue dialog boxes of Figure 2 represents examples of simplified prompts[2]. LLM's high-level responses to these prompts are post-processed and then used as input arguments for the modules to yield low-level specifications of the scene.

The **Floor & Wall Module**, illustrated in the first panel of Figure 2, is responsible for creating floor plans, constructing wall structures, and selecting materials for floors and walls. Each room is represented as a rectangle, defined by four tuples that specify the coordinates of its corners. GPT-4 directly yields the coordinates for placing the rooms and suggests realistic dimensions and connectivity for these rooms. Figure 3 illustrates several examples of diverse layouts this module proposes where HOLODECK generates prompt-appropriate, intricate, multi-room floor plans.

This module also chooses materials for the floors and walls, which is crucial for enhancing the realism of environments. HOLODECK can match LLM proposals to one of 236 materials, each available in 148 colors, enabling semantic customization of scenes. As shown in Figure 4, HOLODECK can generate scenes with suitable materials based on the type of scene, such as opting for concrete walls and floors in a *prison cell* scenario. Inputs with specific texture requirements are often reflected in the final design, for example, "pink color", "red wall bricks," and "checkered floor".

The **Doorway & Window Module**, illustrated in the second panel of Figure 2, is responsible for proposing room connections and windows. Each of these two properties is queried separately from the LLM. The LLM can propose doorways and windows that match 40 door styles and 21 window types, each of which can be modified by several properties, including size, height, quantity, etc. For instance, Figure 5 shows HOLODECK's tailored designs on doors and windows, such as wider doors for "wheelchair accessibility" and multiple floor-to-ceiling windows in a "sunroom" setting.

The **Object Selection Module**, illustrated in the third panel of Figure 2, allows the LLM to propose objects that should be included in the layout. Leveraging the extensive Objaverse asset collection, HOLODECK can fetch and place diverse

[2]The complete prompts (available in the supplementary materials) include additional guidance for LLMs to avoid common errors we observe. For example, by adding a sentence, "the minimal area per room is 9 m^2", HOLODECK can avoid generating overly small rooms.

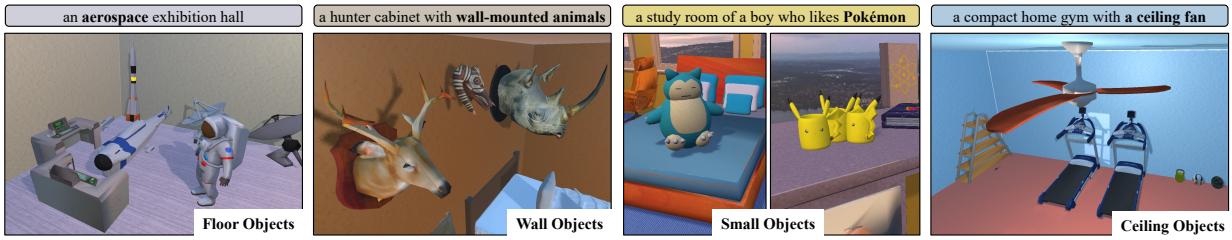

Figure 6. **Objects Customizability.** HOLODECK can select and place appropriate floor/wall/small/ceiling objects conditioned on the input.

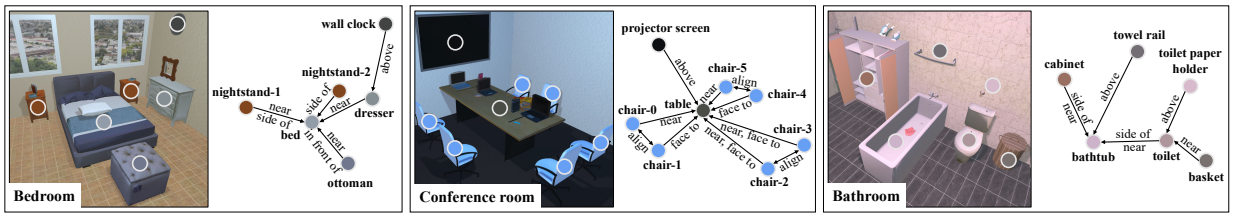

Figure 7. Examples of **Spatial Relational Constraints** generated by LLM and their solutions found by our constraint satisfaction algorithm.

Figure 8. **Output Diversity.** HOLODECK can generate **multiple variants** for the same input with different assets and layouts.

objects in the scene. Queries are constructed with LLM-proposed descriptions and dimensions, like "multi-level cat tower, 60 × 60 × 180 (cm)" to retrieve the optimal asset from Objaverse. The retrieval function[3] considers visual and textual similarity and dimensions to ensure the assets match the design. Figure 6 shows the capability of HOLODECK to customize diverse objects on the floor, walls, on top of other items, and even on the ceiling.

The **Constraint-based Layout Design Module**, illustrated in the fourth panel of Figure 2, generates the positioning and orientation of objects. Previous work [9] shows LLM can directly provide the absolute value of the object's bounding box. However, when attempting to place a diverse lot of assets within environments, this method frequently leads to out-of-boundary errors and object collisions. To address this, instead of letting LLM directly operate on numerical values, we propose a novel constraint-based approach that employs LLM to generate spatial relations between the objects, e.g., "coffee table, in front of, sofa", and optimize the layout based on the constraints. Given the probabilistic nature of LLMs, HOLODECK can yield multiple valid layouts given the same prompt as shown in Figure 8.

Spatial Relational Constraints. We predefined ten types of constraints, organized into five categories: (1) Global: *edge*, *middle*; (2) Distance: *near*, *far*; (3) Position: *in front of*, *side of*, *above*, *on top of*; (4) Alignment: *center aligned* and (5) Rotation: *face to*. LLM selects a subset of constraints for each object, forming a scene graph for the room (examples shown in Figure 7). Those constraints are treated softly, allowing for certain violations when finding a layout to satisfy all constraints is not feasible. Besides those soft constraints, we enforce hard constraints to prevent object collisions and ensure that all objects are within the room's boundaries.

Constraint Satisfaction. We first reformulate the spatial relational constraints defined above into mathematical conditions (e.g., two objects are center-aligned if they share the same x or y coordinate). To find layouts that satisfy constraints sampled by LLMs, we adopt an optimization algorithm to place objects autoregressively. The algorithm first uses LLM to identify an anchor object and then explores placements for the anchor object. Subsequently, it employs Depth-First-Search (DFS)[4] to find valid placements for the remaining objects. A placement is only valid if all the hard constraints are satisfied. For example, in Figure 7, *bed* is selected as the anchor object in the *bedroom*, and the *nightstands* are placed subsequently. The algorithm is executed for a fixed time (30 seconds) to get multiple candidate layouts and return the one that satisfies the most total constraints. We verify the effectiveness of our constraint-based layout in Sec 4.3.

[3]We use CLIP [37] to measure the visual similarity, Sentence-BERT [39] for the textual similarity, and 3D bounding box sizes for the dimension.

[4]Given the linear nature of constraints, a Mixed Integer Linear Programming (MILP) solver can also be employed. While we assume the DFS solver in our experiments, we analyze the MILP solver in the supplements.

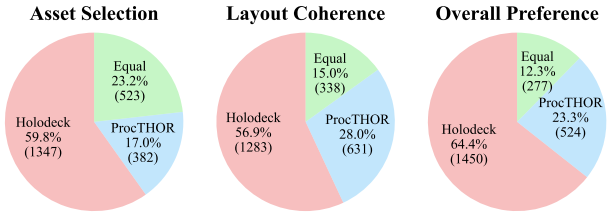

Figure 9. Comparative human evaluation of HOLODECK and PROCTHOR across three criteria. The pie charts show the distribution of annotator preferences, showing both the percentage and the actual number of annotations favoring each system.

Leveraging Objaverse Assets, HOLODECK is able to support the creation of diverse and customized scenes. We curate a subset of assets suitable for indoor design from Objaverse 1.0. These assets are further annotated by GPT-4-Vison [31] automatically with additional details, including textual descriptions, scale, canonical views, etc.[5] Together with the assets from PROCTHOR, our library encompasses 51,464 annotated assets. To import Objaverse assets into AI2-THOR for embodied AI applications, we optimize the assets by reducing mesh counts to minimize the loading time in AI2-THOR, generating visibility points and colliders. More details on importing Objaverse assets into AI2-THOR are available in the supplementary materials.

In the following sections, we will evaluate the quality and utility of the scenes generated by HOLODECK.

4. Human Evaluation

We conduct comprehensive human evaluations to assess the quality of HOLODECK scenes, with a total of 680 graduate students participating in three user studies: (1) a comparative analysis on **residential scenes** with PROCTHOR as the baseline; (2) an examination of HOLODECK's ability in generating **diverse scenes**, and (3) an ablation study to validate the effectiveness of our **layout design** method. Through these user studies, we demonstrate that HOLODECK can create residential scenes of better quality than previous work while being able to extend to a wider diversity of scene types.

4.1. Comparative Analysis on Residential Scenes

This study collects human preference scores to compare HOLODECK with PROCTHOR [6], the sole prior work capable of generating complete, interactable scenes. Our comparison focuses on residential scenes, as PROCTHOR is limited to four types: *bathroom*, *bedroom*, *kitchen*, and *living room*.

Setup. We prepared 120 scenes for human evaluation, comprising 30 scenes per scene type, for both HOLODECK and the PROCTHOR baseline. The PROCTHOR baseline has access to the same set of Objaverse assets as HOLODECK. For HOLODECK, we take the scene type, e.g., "bedroom", as the prompt to generate the scenes. We pair scenes of the same scene type from the two systems, resulting in 120 paired scenes for human evaluation. For each paired scene, we display two shuffled top-down view images of the scenes from the two systems. We ask the annotator to choose which scene is better or equally good based on three questions: (1) **Asset Selection**: which selection of 3D assets is more accurate/faithful to the scene type? (2) **Layout Coherence**: which arrangement of 3D assets adheres better to realism and common sense (considering the position and orientation)? and (3) **Overall Preference**: which of the two scenes would you prefer given the scene type?

[5]GPT-4-Vision can take in multiple images, we prompt it with multi-view screenshots of 3D assets to get the annotations.

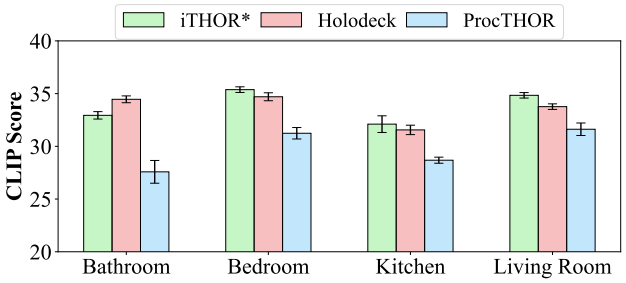

Figure 10. CLIP Score comparison over four residential scene types. * denotes iTHOR scenes are designed by human experts.

Humans prefer HOLODECK over PROCTHOR. Figure 9 presents a clear preference for HOLODECK in the comparative human evaluation against PROCTHOR, with a majority of annotators favoring HOLODECK for Asset Selection (59.8%), Layout Coherence (56.9%), and showing a significant preference in Overall Preference (64.4%).

In addition to human judgments, we employ CLIP Score[6] [15] to quantify the visual coherence between the top-down view of the scene and its corresponding scene type embedded in a prompt template "*a top-down view of [scene type]*". Besides, we add human-designed scenes from iTHOR [23] as the upper bound for reference. Figure 10 shows the CLIP scores of HOLODECK exceed PROCTHOR with great margins and closely approach the performance of iTHOR, demonstrating HOLODECK's ability to generate visually coherent scenes faithful to the designated scene types. The CLIP Score experiment agrees with our human evaluation.

4.2. HOLODECK on Diverse Scenes

To evaluate HOLODECK's capability beyond residential scenes, we have humans rate its performance on 52 scene types[7] from MIT Scenes Dataset [36], covering five categories: Stores (*deli*, *bakery*), Home (*bedroom*, *dining room*), Public Spaces (*museum*, *locker room*), Leisure (*gym*, *casino*) and Working Space (*office*, *meeting room*).

[6]Here, we use OpenCLIP [19] with ViT-L/14 trained on LAION-2B [41]. We use cosine similarity times 100 as the CLIP Score.

[7]Limited by the PROCTHOR framework, we filter those scenes types that require special structures such as *swimming pool*, *subway*, etc.

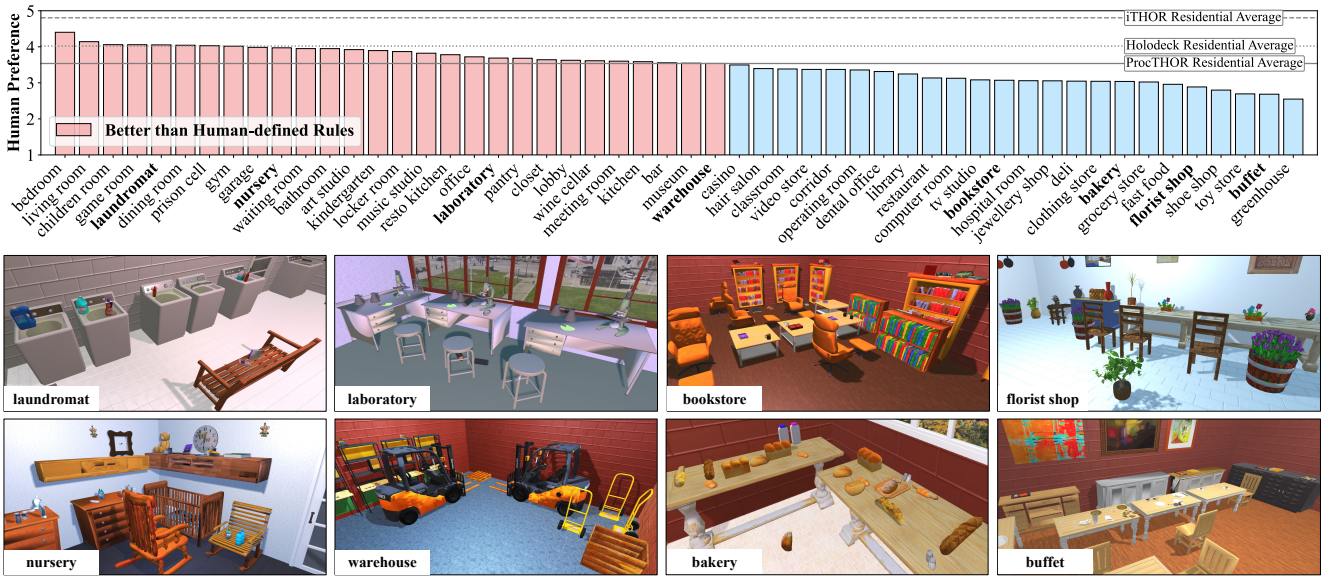

Figure 11. Human evaluation on 52 scene types from MIT Scenes [36] with qualitative examples. The three horizontal lines represent the average score of iTHOR, HOLODECK, and PROCTHOR on four types of residential scenes (*bedroom*, *living room*, *bathroom* and *kitchen*.)

Setup. We prompt HOLODECK to produce five outputs for each type using only the scene name as the input, accumulating 260 examples across the 52 scene types. Annotators are presented with a top-down view image and a 360-degree video for each scene and asked to rate them from 1 to 5 (with higher scores indicating better quality), considering asset selection, layout coherence, and overall match with the scene type. To provide context for these scores, we include residential scenes from PROCTHOR and iTHOR in this study, with 20 scenes from each system.

HOLODECK can generate satisfactory outputs for most scene types. Figure 11 demonstrates the human preference scores for diverse scenes with qualitative examples. Compared to PROCTHOR's performance in residential scenes, HOLODECK achieves higher human preference scores over half of (28 out of 52) the diverse scenes. Given that PROCTHOR relies on human-defined rules and residential scenes are relatively easy to build with common objects and simple layout, HOLODECK's breadth of competence highlights its robustness and flexibility in generating various indoor environments. However, we notice that HOLODECK struggles with scenes requiring more complex layouts such as *restaurant* or unique assets unavailable in Objaverse, e.g., "a dental x-ray machine" for the scene *dental office*. Future work can improve the system by incorporating more assets and introducing more sophisticated layout algorithms.

4.3. Ablation Study on Layout Design

This user study aims to validate the effectiveness of HOLODECK's constraint-based layout design method.

Baselines. We consider four layout design methods: (1) CONSTRAINT: the layout design method of HOLODECK; (2) ABSOLUTE: directly obtaining the absolute coordinates and orientation of each object from LLM akin to LayoutGPT [9]; (3) RANDOM: randomly place all objects in the room without collision; (4) EDGE: placed objects along the walls.

Method	Bathroom	Bedroom	Kitchen	Living Room	Average
ABSOLUTE	0.369	0.343	0.407	0.336	0.364
RANDOM	0.422	0.339	0.367	0.348	0.369
EDGE	0.596	0.657	**0.655**	0.672	0.645
CONSTRAINT	**0.696**	**0.745**	0.654	**0.728**	**0.706**

Table 1. Mean Reciprocal Rank (↑) of different layouts ranked by human. CONSTRAINT: using spatial relational constraints; ABSOLUTE: LLM-defined absolute positions; RANDOM: randomly place the objects and EDGE: put objects at the edge of the room.

Setup. We modify the residential scenes of HOLODECK used in 4.1 by altering the layouts using the previously mentioned methods while keeping the objects in the scene identical. We present humans with four shuffled top-down images from each layout strategy and ask them to rank the four layouts considering out-of-boundary, object collision, reachable space, and layout realism.

Constraint-based layout is more reliable. Table 1 reports the Mean Reciprocal Rank of different layout design methods. HOLODECK's constraint-based approach outperforms the other methods significantly on *bathroom*, *bedroom* and *living room*. CONSTRAINT and EDGE perform similarly on *kitchen*, where it is common to align most objects against walls. The ABSOLUTE method performs no better than RANDOM due to its tendency to create scenes with collision and boundary errors (see examples in the supplement), typically rated poorly by humans. These results endorse spatial relational constraints as a viable strategy for generating scenes that adhere to commonsense logic.

Method	Office		Daycare		Music Room		Gym		Arcade		Average	
	Success	SPL	Success	SPL	Success	SPL	Success	SPL	Success	SPL	Success	SPL
Random	3.90	0.039	4.05	0.041	5.20	0.052	2.84	0.029	2.54	0.025	3.71	0.037
PROCTHOR [6]	8.77	0.031	2.87	0.011	6.17	0.027	0.68	0.002	2.06	0.005	4.11	0.015
+OBJAVERSE (ours)	18.42	0.068	8.99	0.061	25.69	0.157	**18.79**	0.101	**13.21**	**0.076**	17.02	0.093
+HOLODECK (ours)	**25.05**	**0.127**	**15.61**	**0.127**	**31.08**	**0.202**	18.40	**0.110**	11.84	0.069	**20.40**	**0.127**

Table 2. Zero-shot ObjectNav on NOVELTYTHOR. PROCTHOR is the model pretrained on PROCTHOR-10K [6]. +OBJAVERSE and +HOLODECK stand for models finetuned on the corresponding scenes. We report Success (%) and Success weighted by Path Length (SPL).

Figure 12. Zero-shot object navigation in novel scenes. Given a novel scene type, e.g., *Music Room*, HOLODECK can synthesize new scenes for fine-tuning to improve the performance of pretrained agents in expert-designed environments.

5. Object Navigation in Novel Environments

As illustrated in Figure 12, one application of HOLODECK is synthesizing training environments to better match a novel testing distribution. To study this application, we consider ObjectNav [1], a common task in which a robot must navigate toward a specific object category. As existing benchmarks [5, 6, 38] for ObjectNav consider only household environments and support a very limited collection of object types (16 object types in total combining the above benchmarks), we introduce NOVELTYTHOR, an artist-designed benchmark to evaluate embodied agents in diverse environments. Subsequently, we use the ObjectNav model pretrained on PROCTHOR-10K [23] and finetune it on 100 scenes generated by HOLODECK. These scenes are created by prompting HOLODECK with the novel scene type as input. The model is then evaluated on NOVELTYTHOR.

NOVELTYTHOR. We have two professional digital artists manually create 10 novel testing environments with two examples for each of the five categories: *Office*, *Daycare*, *Music Room*, *Gym*, and *Arcade*. Each scene contains novel object types not included in the existing ObjectNav tasks, e.g., "piano" in *Music Room*, "treadmill" in *Gym*, etc. Across NOVELTYTHOR, there are 92 unique object types.

Baselines. For all methods except the one of random action, we use the same pre-trained ObjectNav model from PROCTHOR-10K [23], which has been trained for ≈400M steps to navigate to 16 object categories. To adapt the agent to novel scenes without human-construct training data, we consider two methods: (1) +HOLODECK: we prompt[8] HOLODECK to generate 100 scenes for each scene type automatically; (2) +OBJAVERSE: a strong baseline by enhancing PROCTHOR with HOLODECK's scene-type-specific object selection, specifically, those scenes are populated with similar Objaverse assets chosen by HOLODECK.

Model. Our ObjectNav models use the CLIP-based architectures of [21], which contains a CNN visual encoder and a GRU to capture temporal information. We train each model with 100 scenes for 50M steps, which takes approximately one day on 8 Quadro RTX 8000 GPUs. We select the checkpoint of each model based on the best validation performance on its own validation scenes.

Results. Table 2 shows zero-shot performance on NOVELTYTHOR. HOLODECK achieves the best performance on average and surpasses baselines with considerable margins on *Office*, *Daycare*, and *Music Room*. On *Gym* and *Arcade*, +HOLODECK and +OBJAVERSE perform similarly. Given that the main difference between +HOLODECK and +OBJAVERSE scenes is in the object placements, the observed difference suggests that HOLODECK is more adept at creating layouts that resemble those designed by humans. For example, We can observe in Figure 12 that the music room in NOVELTYTHOR contains a piano, violin cases, and cellos that are in close proximity to each other. The music room generated by HOLODECK also shows a similar arrangement of these objects, highlighting the "commonsense" understanding of our method. PROCTHOR struggles in NOVELTYTHOR, often indistinguishably from random, because of poor object coverage during training.

6. Conclusion and Limitation

We propose HOLODECK, a system guided by large language models to generate diverse and interactive Embodied AI environments with text descriptions. We assess the quality of HOLODECK with large-scale human evaluation and validate its utility in Embodied AI through object navigation in novel scenes. We plan to add more 3D assets to HOLODECK and explore its broader applications in Embodied AI in the future.

[8] Here, we prompt with the scene name and its paraphrases to get more diverse outputs, e.g., we use "game room", "amusement center" for *Arcade*.

References

[1] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects. In *arXiv:2006.13171*, 2020. 8

[2] Angel Chang, Manolis Savva, and Christopher D Manning. Interactive learning of spatial knowledge for text to 3d scene generation. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 14–21, 2014. 2

[3] Angel X Chang, Mihail Eric, Manolis Savva, and Christopher D Manning. Sceneseer: 3d scene design with natural language. *arXiv preprint arXiv:1703.00050*, 2017. 2

[4] Yu Cheng, Yan Shi, Zhiyong Sun, Dezhi Feng, and Lixin Dong. An interactive scene generation using natural language. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6957–6963. IEEE, 2019. 2

[5] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3164–3174, 2020. 2, 8

[6] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In *NeurIPS*, 2022. Outstanding Paper Award. 2, 3, 6, 8

[7] Matt Deitke, Rose Hendrix, Ali Farhadi, Kiana Ehsani, and Aniruddha Kembhavi. Phone2proc: Bringing robust robots into our chaotic world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9665–9675, 2023. 2

[8] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 2, 3

[9] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *arXiv preprint arXiv:2305.15393*, 2023. 3, 5, 7

[10] Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel. Scenescape: Text-driven consistent scene generation. *arXiv preprint arXiv:2302.01133*, 2023. 2, 3

[11] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021. 3

[12] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020. 2

[13] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In *International Conference on Machine Learning*, pages 11808–11826. PMLR, 2023. 3

[14] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9984–9993, 2019. 3

[15] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 6

[16] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. *arXiv preprint arXiv:2303.11989*, 2023. 2, 3

[17] Ruizhen Hu, Zeyu Huang, Yuhan Tang, Oliver Van Kaick, Hao Zhang, and Hui Huang. Graph2plan: Learning floorplan generation from layout graphs. *ACM Transactions on Graphics (TOG)*, 39(4):118–1, 2020. 2

[18] Ian Huang, Vrishab Krishna, Omoruyi Atekha, and Leonidas Guibas. Aladdin: Zero-shot hallucination of stylized 3d assets from abstract scene descriptions. *arXiv preprint arXiv:2306.06212*, 2023. 3

[19] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 6

[20] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, S. Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5:6670–6677, 2019. 2

[21] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8

[22] 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. *arXiv preprint*, 2023. 2

[23] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017. 2, 3, 6, 8

[24] 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 ev-

eryday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023. 2

[25] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 3

[26] Yiqi Lin, Hao Wu, Ruichen Wang, Haonan Lu, Xiaodong Lin, Hui Xiong, and Lin Wang. Towards language-guided interactive 3d generation: Llms as layout interpreter with generative feedback. *arXiv preprint arXiv:2305.15808*, 2023. 3

[27] Rui Ma, Akshay Gadi Patil, Matthew Fisher, Manyi Li, Sören Pirk, Binh-Son Hua, Sai-Kit Yeung, Xin Tong, Leonidas Guibas, and Hao Zhang. Language-driven synthesis of 3d scenes from scene databases. *ACM Transactions on Graphics (TOG)*, 37(6):1–16, 2018. 2

[28] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023. 3

[29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[30] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019. 3

[31] OpenAI. GPT-4V(ision) System Card, 2023. 6

[32] R OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023. 2

[33] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in Neural Information Processing Systems*, 34:12013–12026, 2021. 2

[34] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 3

[35] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8494–8502, 2018. 2

[36] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 413–420. IEEE, 2009. 6, 7

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 5

[38] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 2, 8

[39] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019. 5

[40] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019. 2

[41] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 6

[42] Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5466–5475, 2023. 2

[43] 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. 2

[44] Fuwen Tan, Song Feng, and Vicente Ordonez. Text2scene: Generating compositional scenes from textual descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6710–6719, 2019. 2

[45] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Scene graph denoising diffusion probabilistic model for generative indoor scene synthesis. *arXiv preprint arXiv:2303.14207*, 2023.

[46] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. In *2021 International Conference on 3D Vision (3DV)*, pages 106–115. IEEE, 2021. 2

[47] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *arXiv preprint arXiv:2311.01455*, 2023. 2

[48] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma,

Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022. 2

[49] Qiuhong Anna Wei, Sijie Ding, Jeong Joon Park, Rahul Sajnani, Adrien Poulenard, Srinath Sridhar, and Leonidas Guibas. Lego-net: Learning regular rearrangements of objects in rooms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19037–19047, 2023. 2

[50] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016. 3

[51] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018. 2

[52] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019. 3

[53] Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. Text2nerf: Text-driven 3d scene generation with neural radiance fields. *arXiv preprint arXiv:2305.11588*, 2023. 2, 3

[54] Yiqun Zhao, Zibo Zhao, Jing Li, Sixun Dong, and Shenghua Gao. Roomdesigner: Encoding anchor-latents for style-consistent and shape-compatible indoor scene generation. *arXiv preprint arXiv:2310.10027*, 2023. 2

[55] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 3