This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

SAGE: Scalable Agentic 3D Scene Generation for Embodied AI

Hongchi Xia[1,2*] Xuan Li[1] Zhaoshuo Li[1] Qianli Ma[1] Jiashu Xu[1] Ming-Yu Liu[1] Yin Cui[1]
Tsung-Yi Lin[1] Wei-Chiu Ma[3] Shenlong Wang[2] Shuran Song[1,4] Fangyin Wei[1]

[1]NVIDIA [2]University of Illinois Urbana-Champaign
[3]Cornell University [4]Stanford University

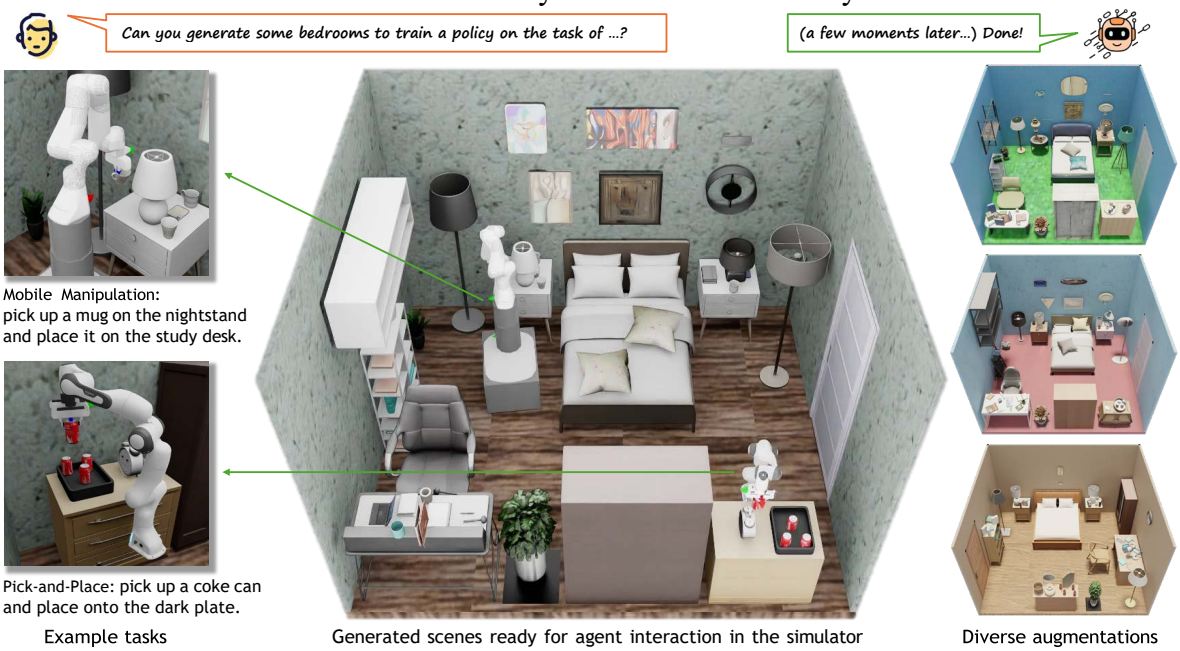


Figure 1. **Overview and example outputs of SAGE**. Given an open-ended user request, our system autonomously composes realistic, diverse, and simulation-ready 3D environments. The generated scenes are directly deployable in modern simulators, supporting embodied tasks such as Mobile Manipulation and Pick-and-Place. Through agent-driven reasoning, generator orchestration, and multi-level augmentation, the framework produces interactive environments at scale for robot policy learning.

Abstract

Real-world data collection for embodied agents remains costly and unsafe, calling for scalable, realistic, and simulator-ready 3D environments. However, existing scene-generation systems often rely on rule-based or task-specific pipelines, yielding artifacts and physically invalid scenes.

*We present SAGE, an agentic framework that, given a user-specified embodied task (*e.g.*, "pick up a bowl and place it on the table"), understands the intent and automatically generates simulation-ready environments at scale. The agent couples multiple generators for layout and object composition with critics that evaluate semantic plausibility, visual realism, and physical stability. Through iterative reasoning and adaptive tool selection, it self-refines the scenes until meeting user intent and physical validity.*

The resulting environments are realistic, diverse, and directly deployable in modern simulators for policy training. Policies trained purely on this data exhibit clear scaling trends and generalize to unseen objects and layouts, demonstrating the promise of simulation-driven scaling for embodied AI. Code, demos, and the SAGE-10k dataset can be found on the project page here.

1. Introduction

Embodied AI is on the hunt for data. Unfortunately, while the web has powered large vision and language models [2, 37], neither the web nor the real world can resolve the pressing data needs of embodied agents. Real-world embodied data collection is slow and costly [50], and is fundamentally constrained by the need for *interactive environments* [39]. Simulation thus emerges as the natural alternative, providing scalable, interactive, and low-cost embodied envi-

*This work was done while Hongchi Xia was an intern at NVIDIA.

Framework	Method	Input		Methodology				Simulation-Ready Output		
		Open-Vocabulary	Modality	Built-in Scene Generation	Grounded in 3D	Self-Improvement	Fine-grained Control	Collision Check	w/ Physical Attributes	Simulator-Validated
Rule-based	Infinigen [38], ProcTHOR [7]	✗	Text	✓	✓	✗	✗	✗	✗	✗
3D Data-Driven	DiffuScene [44], Text2Room [15]	✗	Text / 3D Scene	✓	✓	✗	✗	✗	✗	✗
	ATISS [35], PhyScene [59]		Floor Plan							
	CommonScenes [64], EchoScene [65]		Scene Graph							
LLM/VFMs Static Pipeline	LayoutGPT [9], LayoutVLM [41], I-Design [3]	✓	Text	✓	✗	✗	✗	✗	✗	✗
	Holodeck [60], ArtiScene [14], Architect [52]				✓	✓(pose-only)		✓		
	AnyHome [10], Scenethesis [24], GenUSD [23]									
Agent-based	SceneWeaver [61]	✓	Text	✗	✓	✓	✓	✓	✗	✗
	SAGE (Ours)		Text (+Image)	✓					✓	✓

Table 1. **Comparison of scene generation methods.** Our agent-based method uniquely fulfills all criteria, enabling scalable generation of simulator-validated data essential for robotic applications.

ronments with safety guarantees [28].

To effectively support embodied AI, simulated data must satisfy four desiderata: (i) **Realism**: the geometry, appearance, semantic structure, and physics of the simulation should resemble the real world closely enough that policies learned in simulation can reliably transfer to reality; (ii) **Diversity**: simulation should encompass a wide range of assets, environments, and tasks to prevent overfitting and support generalization; (iii) **Simulation-readiness**: objects and scenes must be physically stable, interactable, and directly compatible with modern simulators for large-scale training; (iv) **Task-awareness**: simulation environments should adapt to and facilitate the training of various targeted embodied tasks. For instance, a data engine should provide various kitchens for household robots, fire-hazard scenes for rescue robots, surgical rooms for medical robots, *etc*.

Unfortunately, existing methods for generating 3D simulated data fall short on one or more of these criteria. While Real2Sim approaches [17, 54–56] prioritize realism by reconstructing digital twins from the real world, the high costs of data capture and reconstruction make them difficult to scale. In contrast, generative approaches offer a more scalable alternative, but still face significant bottlenecks. For example, rule-based systems [7, 38] ensure physical plausibility but sacrifice flexibility and diversity. Data-driven approaches [15, 35, 44] improve realism, but their limited 3D training data prevents them from generalizing to new room types, handling open-vocabulary prompts, or supporting fine-grained layout control. Pipelines based on foundation models [9, 41, 60] enable text-driven generation but lack 3D grounding, yielding physically invalid scenes. Perhaps most importantly, these systems are *static*: their "computational graph" is fixed, preventing adaptive reasoning and self-correction. Recently, a concurrent work, SceneWeaver [61], takes a step toward agentic scene generation. Yet, it is not directly deployable in simulation due to missing physical attributes and the absence of simulator-in-the-loop verification. There are still major gaps in (1) physical grounding for interaction and (2) compatibility with robot simulators, as shown in Tab. 1.

In this work, we present SAGE, a novel agentic framework for scalable 3D scene generation that produces simulation-ready environments directly from arbitrary user prompts. The agent, operating over Model Context Protocal (MCP) [29], adaptively orchestrates generators for floor plans, structured layouts, and text-to-3D assets, while two complementary critics provide continuous feedback: a visual critic for semantic/spatial coherence and a physics critic with simulator-in-the-loop validation (Isaac Sim [32]) for stability under gravity and collisions. This closed loop enables self-correction without hard-coded tool order and yields scenes that are both realistic and interactable.

We scale these scenes for embodied AI by applying multi-level augmentation (object configuration, object category, and layout) to generate diverse yet task-consistent environments. On top of these scenes, we automatically synthesize action data with grasp pose proposals [63], collision-aware Inverse Kinematics (IK) [42], and navigation planning, and train a Diffusion Policy [4]. Experiments show clear scaling trends with increasing scene diversity and demonstration count, and improved generalization to unseen objects and layouts.

In summary, our contributions are twofold:

- **An agentic scene-generation framework** that unifies multiple generators with visual and physics critics under MCP, enabling adaptive tool use, self-improvement, and simulator-validated stability. It generates open-vocabulary scenes with state-of-the-art realism and physical validity.
- **An embodied AI data pipeline** that automatically generates unlimited scenes and teacher demonstrations for imitation learning, yielding clear scaling benefits and stronger generalization to unseen environments.

2. Related Works

3D Indoor Scene Synthesis Early work synthesized indoor scenes with procedural systems such as ProcTHOR [7] and Infinigen Indoors [38], assembling layouts via rules and grammars. This paradigm grounds geometry and scales well but fixed recipes constrain open vocabulary, fine-grained

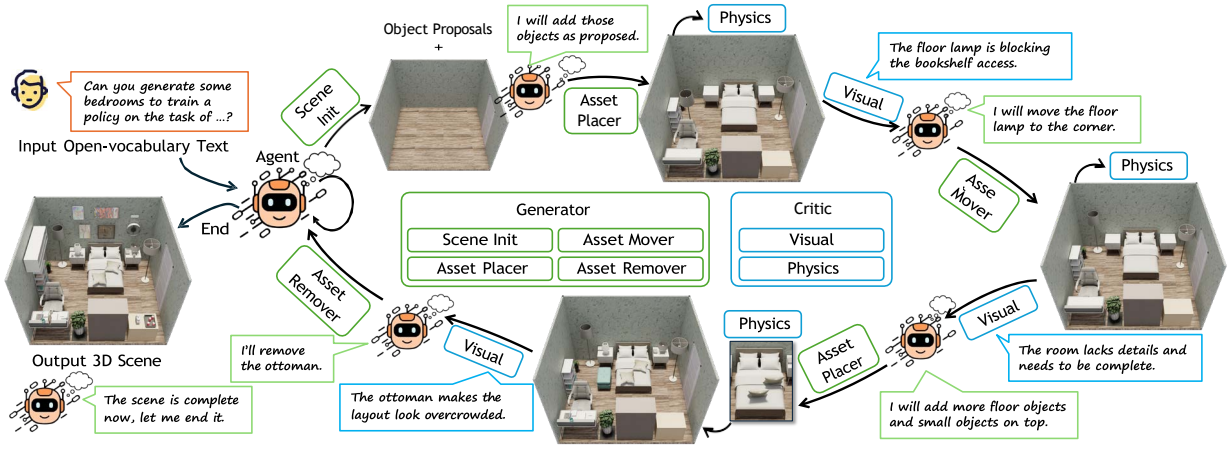


Figure 2. **Overview of SAGE scene generation.** Our system converts open-vocabulary text prompts into simulation-ready 3D scenes by orchestrating multiple generator tools and critics. The agent dynamically calls generators (Scene Init, Asset Placer/Mover/Remover) to construct and refine layouts, while visual and physics critics provide iterative feedback for self-improvement. The visual critic suggests semantic corrections (*e.g.*, missing or misplaced objects), and the physics critic validates stability via Isaac Sim. For example, after applying physics critic in the bottom image, the newly added pillows on the bed fall flat. This self-improvement process ends when the agent considers that the generated scene meets the input user requirements. The resulting scenes can be further scaled via augmentation and used for embodied policy learning.

control, and self-improvement, and rarely expose physics-rich attributes beyond hand-coded rules. Data-driven methods such as ATISS [35] and DiffuScene [44], along with graph-aware CommonScenes [64] and EchoScene [65], learn strong spatial priors and yield realistic layouts with neural 3D generators; nevertheless they inherit closed taxonomies, offer limited object-level levers, and do not attach explicit physical properties or run simulator-side validation. Motivated by advances in long-horizon reasoning for LLM agents [12, 53, 62], recent methods shift to language- or vision-assisted scene synthesis. Holodeck [60] and ArtiScene [14] leverage LLMs or image intermediaries to widen semantic coverage and stylistic control; however, they usually follow static, pre-ordered modules, provide little self-improvement, have sparse 3D grounding, and omit physics validation such as collision or stability. Some works [10, 24] combine LLM planning with vision-guided refinement under constraints, but the refinement is only on asset pose. Building further on agent capabilities for iterative decision making and self-reflection [16, 40], agentic pipelines [23, 61] generate complete layouts and refine them through feedback. SceneWeaver [61] further leverages tool use for relation/collision-aware placement. Despite these advances, orchestration often outweighs end-to-end simulation deliverables: physical attributes are not systematically attached and simulator-validated outputs are not the default artifact. Our method SAGE pushes from "semantically plausible" to task-aligned, simulation-ready: we support open-vocabulary prompts (with optional images), generate scenes natively, attach object-level physical properties, and perform in-simulator validation (collision and stability), so outputs are directly deployable for embodied policy training.

Simulation Environment for Embodied AI In recent years, numerous simulation benchmark environments have been developed to provide safe, scalable, and controllable platforms for evaluating robot learning algorithms before real-world deployment. LIBERO [25], CALVIN [27] and SimplerEnv [22] focus on tabletop manipulation tasks such as pick-and-place and object rearrangement using simple robotic arm setups. iGibson [20], Habitat [43], BEHAVIOR-1K [21], ManiSkill [45], RoboCasa [30], AI2Thor [18], VirtualHome [36], ThreeDWorld [11] extend the scope to home-scale embodied activities that combine object interaction and motion planning within complex 3D indoor scenes. Some recent works also aim to unify and scale up embodied AI benchmarking. RoboGen [51] combines automatic object generation and task generation. RoboVerse [13] integrates multiple simulators and benchmarks to support seamless transitions across environments. Underlying these platforms are diverse physics engines such as PyBullet [5], MuJoCo [47], Isaac Sim [32], SAPIEN [57], Genesis [1], which are specifically developed for robotics simulation, and Unity [48], Unreal [8], which are designed for game developments. While many of these frameworks provide rich, realistic, and ready-to-use environments for embodied AI, they often require substantial manual effort for scene and asset creation. Others rely on procedurally retrieved assets, which can limit diversity when scaling up. And the simulation is often only enabled when the creation is complete. In contrast, SAGE integrates 3D generation frameworks, enabling open-vocabulary asset creation with high flexibility and minimal manual intervention. Furthermore, by incorporating simulation directly into the generation loop, SAGE supports iterative self correction and improvement.

3. Method

Given a user demand with a robot task description, our goal is to generate diverse 3D scenes ready to run embodied agents in simulation for scalable policy learning. At the core of SAGE is an agent-driven scene generation framework, detailed in Sec. 3.1. Based on the feedback from visual and physical *critics*, the agent effectively self-improves the generation with multiple editing operations supported by scene *generator* tools. In Sec. 3.2, we describe how scene generation can be easily scaled up with object-level and scene-level augmentation, which can then be used for action generation to train embodied AI policy.

3.1. Agent-driven Scene Generation

SAGE operates under the Model Context Protocol (MCP), a standardized protocol for seamless interaction with external tools [29]. In this setup, the agent acts as the MCP client, while each tool (*e.g.*, layout generator, physics simulator) is hosted behind an MCP server. The agent takes a chain of actions to improve the scene until the scene is considered visually realistic and physically stable. In each iteration, when the agent identifies the need for a specific capability, *e.g.*, generating a floor plan or validating physical stability, it sends a structured request through MCP. The server executes the corresponding operation, returns the result, which the agent incorporates as feedback to decide the next action. This setup enables adaptive, tool-driven scene generation without hard-coded logic. Fig. 2 summarizes our approach.

3.1.1. Generator

The scene is constructed through a set of generator tools that the agent dynamically invokes via MCP. Each tool performs a specific operation – initializing layouts, adding new assets, or adjusting existing ones – and can be flexibly composed based on the agent's reasoning and critic feedback. This allows for iterative, adaptive scene construction with fine-grained control over content and structure.

Scene Initializer takes the scene specification as input and is responsible for generating an empty 3D room (with only floor and walls). It also outputs a list of proposed objects, each with a text description, estimated physical attributes, and placement constraints (*e.g.*, relationship to other objects, boundaries). The textures of floor and wall are generated by MatFuse [49], with a size assigned from the LLM prediction. The object list reflects the scene demands or robot task descriptions. For example, if the user asks about learning a task of "pick an apple and place it to a bowl", the generator will include the required apple and bowl in the object list to be returned to the agent.

Asset Placer takes in a text string describing the placement requirement for a few objects. Its task is to generate and place these objects into the 3D scene. We generate objects with TRELLIS [58] given a text description. To make the object simulation-ready, we also leverage a VLM to estimate physical properties, including height to rescale the unit-sized object, mass for physical simulation, and metallic/roughness values for physics-based rendering (PBR). For each object, we use an LLM to analyze the input placement condition and choose one of the three categories: floor, wall, and on-top. This further guides the placement sequence and constraints of each placement. We adopt depth first search with collision avoidance to place the objects to possible location candidates following the sequence, and choose the placement the best satisfies the constraints.

Asset Mover locates and moves the object specified in the input text string. This is achieved by first removing the object and then reusing the placement planner from the Asset Placer to reloate the object. Movement instructions typically come from the visual critic.

Asset Remover removes the object described in the input text, using LLM-based reasoning to locate the object in the scene. It is usually called when the critics (described later) gives feedback to remove an object.

3.1.2. Critic for Self-Improvement

Naively stacking different generators causes error accumulation from individual imperfections. Errors generally fall into two types: visual artifacts (*e.g.*, missing or misplaced objects) and physical violations (*e.g.*, instability or collisions). To mitigate this, we introduce a visual critic that assesses semantic and spatial coherence, as well as a physics critic that enforces stability and simulator readiness. Together, they guide iterative self-correction high-quality scene generation.

Visual Critic To evaluate and improve the layout quality and scene completeness, we introduce a visual critic. The critic takes the current scene configuration as input, *i.e.*, object placements and multi-view renderings (top-down and four corner views), and proposes new objects to place as well as adjustments to existing placements. Its feedback is then incorporated to aid the agent's decision-making of the most appropriate generator to invoke next. By integrating these visual feedback signals, the agent gains a more comprehensive understanding of the current scene state and can decide which generator to invoke next for optimal refinement.

Physics Critic Beyond visual realism, a generated scene must also be physically stable and simulation-ready to support embodied learning. To achieve this, we employ simulation-in-the-loop validation during generation. After each object addition, movement, or removal. the scene is loaded into Isaac Sim [32] to test its physical stability. We measure the pose change after simulation and reject placements that cause instability or collisions, retaining only candidates that preserve global stability. If the generator fails to

find a stable configuration, the critic reports the failure to the agent, suggesting alternative actions such as using smaller objects or adjusting placement target locations. Through this iterative validation loop, SAGE maintains near-perfect physical stability, ensuring that the resulting scenes are directly deployable for downstream embodied learning tasks.

3.2. Scaling the Scene for Embodied AI

To train a robust and generalizable policy, it is not sufficient to use a single generated scene; instead, we must generate a diverse set of simulation-ready scenes that follow the user's task specification. To this end, we introduce a scene augmentation strategy that systematically expands each generated environment into numerous variants while preserving task semantics. These augmented scenes are then used to generate corresponding robot actions via motion planning, followed by imitation learning to train task-specific embodied policies.

3.2.1. Scene Augmentation

To scale one generated scene into a diverse yet task-consistent set of variants, we apply (1) task-relevant object augmentation (configuration, category) that diversifies key objects while preserving task semantics, and (2) task-irrelevant scene augmentation to enrich the rest of the scenes.

Object Configuration-level The pose of each task-relevant object (*e.g.*, the target to be picked, placed on, or navigated toward) is resampled within the current scene to create variations in object placement. This process increases spatial diversity while preserving overall scene semantics.

Object Category-level Given the text description of each task-relevant object from the generation stage, we employ an LLM-based text augmentation to produce variations in geometry and texture (*e.g.*, shape, color, material, or finish) while maintaining the original object category. We then use TRELLIS [58] to synthesize corresponding 3D assets from these augmented descriptions, which are placed into the scene to enrich visual and physical diversity across instances.

Scene Layout-level While the above augmentations modify task-related objects, the background environment remains unchanged. For tasks requiring full-scene exploration or navigation, we introduce layout-level augmentation, where the background scene, including room geometry and all task-irrelevant objects, is regenerated through the agent-driven scene generation. This process produces diverse scene layouts sharing the same task specification, enabling learning policies that generalize across spatial configurations.

Simulation-ready Validation After each augmentation step, we call the physics critic to ensure the stability and physical plausibility of all placements. Object-level physical properties such as mass and PBR parameters are estimated by a VLM as before. This guarantees that every augmented scene remains physically valid and immediately deployable for policy training.

3.2.2. Action Generation

The generated environments are now ready for downstream embodied AI tasks. We demonstrate their utility on two representative tasks, *i.e.*, Pick-and-Place and Mobile Manipulation, and automatically generate large-scale action demonstrations using established motion planning techniques. This enables scalable policy learning and highlights the benefits of simulation-ready scene generation.

Pick-and-Place For grasping actions, we use M2T2 [63] to generate grasp pose candidates from rendered depth images. Collision-free trajectories are computed by integrating Curobo [42] into the motion planning and inverse kinematics pipeline, ensuring feasible and stable grasp execution.

Mobile Manipulation This task is composed of navigation with object pick-and-place in between. For the navigation motions, we adopt RRT [19] for robot path planning, generating collision-free trajectories between designated start and target positions.

Failure Modes Motion planning is not always successful, often due to inaccurate grasp pose predictions, unreachable target configurations, or unexpected collisions during trajectory execution. We filter out failed examples by performing collision checks and verifying whether each manipulated object reaches its expected location.

3.2.3. Policy Learning

Given the large-scale action data generated from motion planning, we employ imitation learning to train generalizable policies from these demonstrations. Specifically, we use Diffusion Policy [4] for policy learning. The model takes as input RGB and depth images from multiple camera views along with corresponding end-effector trajectories, and outputs continuous actions for next-step execution.

4. Experiments

We evaluate scene generation in Sec. 4.1. In Sec. 4.2, we demonstrate how scalable scene and action data generated by our framework improves policy generalization.

4.1. Scene Generation

4.1.1. Setup

Implementation Details SAGE integrates a series of large foundation models, including LLMs, VLMs, and specialized generators for 3D objects and background textures. We use open-source models hosted via self-managed APIs to ensure reproducibility. Specifically, we adopt gpt-oss-120b [34] as both the agent LLM and the integrated LLM, and Qwen3-VL-30B-A3B-Instruct [46] for vision–language reasoning.



Figure 3. **Common and open-vocabulary scene generation comparison.** Compared with baselines, SAGE produces more complete scenes with more realistic layouts on common room types, while following the style prompts more faithfully on open-vocabulary queries.

Room Type	Method	Visual					Physics	
		#Obj ↑	Real. ↑	Func. ↑	Lay.↑	Comp. ↑	Coll. % ↓	Stab. % ↑
Bedroom	Holodeck[60]	28.5	7.4	6.8	5.0	6.1	29.1	51.0
	SceneWeaver[61]	17.5	9.0	9.7	7.8	7.5	31.0	58.8
	SAGE (Ours)	**48.3**	**9.0**	**10.0**	**8.0**	**9.5**	**2.3**	**99.8**
Kitchen	Holodeck[60]	28.5	6.7	6.1	4.4	6.2	16.0	73.8
	SceneWeaver[61]	37.5	8.2	7.7	6.8	7.2	28.0	67.0
	SAGE (Ours)	**47.6**	**8.5**	**9.0**	**7.8**	**7.6**	**0.7**	**100.0**
Living Room	Holodeck[60]	34.0	8.3	7.3	5.7	7.3	20.8	66.5
	SceneWeaver[61]	18.1	8.5	9.3	7.2	6.8	39.5	77.2
	SAGE (Ours)	**48.8**	**8.8**	**9.5**	**7.8**	**7.4**	**2.7**	**100.0**
Average	Holodeck[60]	30.3	7.5	6.7	5.0	6.5	22.0	63.8
	SceneWeaver[61]	24.4	8.6	8.9	7.3	7.2	32.8	67.7
	SAGE (Ours)	**48.2**	**8.8**	**9.5**	**7.9**	**8.2**	**1.9**	**99.9**

Table 2. **Scene generation evaluation on common scene types.** Scores averaged across 10 scenes per room type. SAGE consistently outperforms prior methods across all categories.

Metrics Following [61], we evaluate the visual quality and physical plausibility with scores averaged across 10 generated scenes per room type. Visual metrics cover Realism, Functionality, Layout, and Completeness based on GPT-4.1 [33]. Physical validity includes the collision ratio of 3D object meshes using trimesh [6], and the ratio of stable objects within Isaac Sim [32]. An object is considered unstable if its relative translation exceeds 0.2 meter or its rotation exceeds 8 degrees after 120 simulation steps.

Baselines We compare against two state-of-the-art methods using their official codebases. **Holodeck** [60] is LLM-driven but lacks self-improvement due to its fixed generation pipeline. **SceneWeaver** [61] is agent-based yet omits simulator validation, yielding non–simulation-ready scenes.

4.1.2. Experiment Results

Common Types We evaluate SAGE and the baselines on three common indoor scene types: Bedroom, Kitchen, and Living Room. Quantitative results are reported in Tab. 2, with qualitative comparisons in Fig. 3. SAGE achieves the best results across all metrics, in both visual quality and physical stability. Holodeck [60] generates fewer objects with lower realism and functionality, and exhibits frequent collisions due to its rigid, predefined generation pipeline. SceneWeaver [61] achieves moderate visual quality but exhibits high collision rates and low stability, mainly due to its absence of simulator-based validation. As illustrated in Fig. 5, baseline scenes show displaced or fallen objects after simulation, whereas SAGE remains fully stable.

Open-Vocabulary Types We further demonstrate the open-vocabulary generation capability of SAGE across a wide range of scene types. As shown in Fig. 3 and Fig. 4, our method successfully generates highly diverse and stylized spaces (*e.g.*, Gym, Office, Cyberpunk game den, Starry-night bedroom). Unlike retrieval-based methods, our text-to-3D object synthesis enables the creation of long-tail, semantically coherent scenes with faithful adherence to user prompts and distinctive visual styles.

Ablation Study We conduct an ablation study to assess the impact of each critic design. As shown in Tab. 3, adding the visual critic substantially improves visual quality, while the physics critic greatly reduces collisions from 7.8% to 1.9% and raises stability to 99.6%. Combining both critics yields the best overall performance across all metrics, confirming that visual feedback and simulation-in-the-loop validation



Figure 4. **Additional open-vocabulary generation.** SAGE produces diverse, semantically coherent scenes spanning various styles and functionalities, from Gym and Office spaces to creative themes like "Cyberpunk game den" and "Starry-night bedroom".

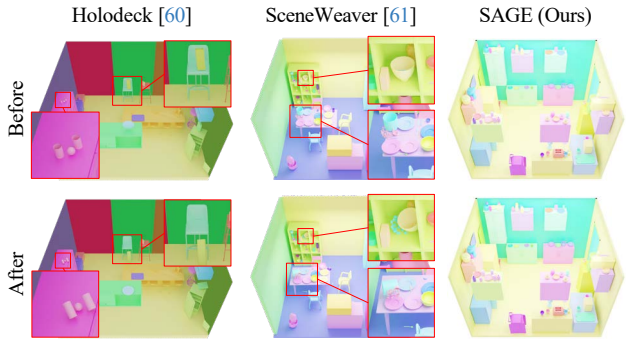


Figure 5. **Stability verification.** Generated scenes are loaded into IsaacSim for physical validation. Both baselines exhibit displaced objects due to instability, whereas SAGE preserves scene stability before and after simulation.

Critics Setting		Visual					Physics	
Visual	Physics	#Obj ↑	Real. ↑	Func. ↑	Lay.↑	Comp. ↑	Coll. % ↓	Stab. % ↑
✗	✗	35.3	8.5	9.2	7.5	7.3	7.8	80.3
✓	✗	<u>50.1</u>	**8.9**	<u>9.5</u>	**8.1**	<u>8.1</u>	3.7	84.1
✗	✓	36.8	8.8	9.3	7.7	7.8	<u>1.9</u>	<u>99.6</u>
✓	✓	**53.7**	**8.9**	**9.6**	<u>8.0</u>	**8.2**	**0.8**	**100.0**

Table 3. **Ablation study on critics.** Results are averaged over five scenes for each of three room types. Adding both visual and physics critics leads to the best results, confirming their joint importance in scene generation. Best is in **bold**, second best is <u>underlined</u>.

are complementary and critical for generating realistic and physically stable scenes.

4.2. Embodied AI Learning

We evaluate whether the diversity of our generated scenes enables effective scaling by measuring policy improvements under increasing scene and demonstration counts.

4.2.1. Setup

Implementation Details We study two representative tasks: (1) Pick-and-Place with a Franka Emika Panda robot, (2) Mobile Manipulation using a Franka Emika Panda arm mounted on an Omron LD-60 mobile base [31, 66]. All action data generation and policy rollout are conducted in Isaac Lab [28] on NVIDIA L40S GPUs with parallel simulation. Policies are trained via the imitation learning framework Robomimic [26]. Pick-and-Place uses a single model, whereas the long-horizon Mobile Manipulation is decomposed into four sequential policies. Each policy is responsible for one stage of the task (see task description) and also predicts a termination signal for stage transition. We report success rates averaged across three held-out scenes, each evaluated with 90 random robot spawning poses and object configurations.

Task Description For Pick-and-Place, the robot must pick up a mug from the table and place it into a bowl. For Mobile Manipulation, the robot is initialized at a random position in the scene. It needs to navigate to a table with a coke can, pick it up, move to another desk, and place the can on it.

Scene and Action Generation SAGE can automatically generate large-scale scene and action data starting from the task descriptions. Object-level and scene-level augmentation is applied to Pick-and-Place and Mobile Manipulation, respectively. We collect over 28k demonstrations for Pick-and-Place with 264 unique objects and nearly 50k demonstrations for Mobile Manipulation across 50 diverse scenes.

4.2.2. Policy Learning

Scaling Trend with SAGE-generated Data As shown in Fig. 6 and Fig. 7, policy success rates increase with the number of scenes. Despite that the learned policy only has access to partial visual observations, it gradually converges towards the privileged agent which has full access to the entire 3D scene information. To contextualize performance relative to task difficulty, we also compare the success rate between policy rollout and motion planning in Tab. 4. Our results show that the learned policy achieves success rates approaching those of the underlying motion planner, confirming both the fidelity of the generated data and the effectiveness of imitation learning at scale.

Comparison with Baselines Since existing methods rely on heterogeneous simulation setups and implementations, directly integrating them into a unified environment is nontrivial. To enable fair comparison, we design two baselines that mimic the methodology of prior works by removing components unique to our framework and substituting them with their counterparts. As shown in Fig. 7, both baselines improve slowly with scale–achieving less than one-third of the

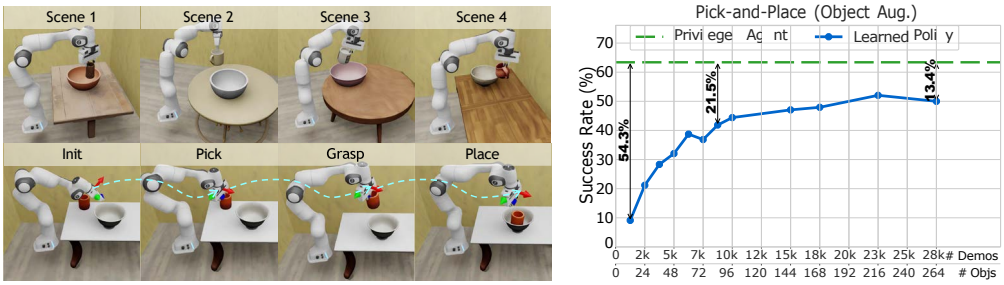


Figure 6. **Examples and scaling curve on Pick-and-Place.** *Top left*: diverse generation. *Bottom left*: example trajectory. *Right*: success rate *w.r.t.* demo/object counts. More diverse object augmentations improve policy success, narrowing the gap to the privileged agent.

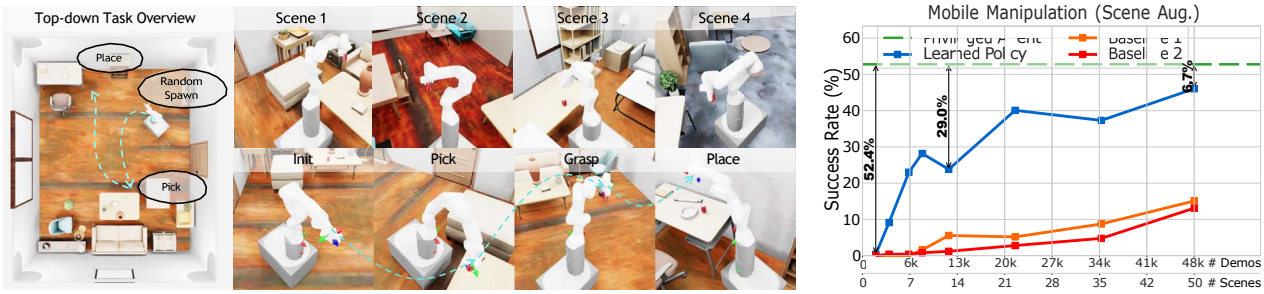


Figure 7. **Examples and scaling curve on Mobile Manipulation.** *Left*: task overview. *Mid-top*: diverse generation. *Mid-bottom*: example trajectory. *Right*: success rate *w.r.t.* demo/scene counts. Both baselines omit physics critic and replace text-to-3D object synthesis with retrieval: **Baseline 1** mimics SceneWeaver [61]; **Baseline 2** further replaces the agent with a fixed pipeline, resembling Holodeck [60]. Diverse SAGE-augmented scenes boost the **learned policy**'s success and close the gap to the privileged agent, while removing physics critic and object synthesis degrades performance (**Baseline 1**). Replacing the agent with a static pipeline further reduces success rate (**Baseline 2**).

Task	Privileged Agent		Policy Rollout	
	Train	Test	Train	Test
Pick-and-Place	65.3	57.7	63.4	50.0
Mobile Manipulation	68.4	52.8	54.6	46.0

Table 4. **Comparison of training and test success rates for policy rollout and motion planning.** The learned policy generalizes well from training to testing, and policy rollout performance approaches that of the privileged motion planning.

Training Source	Test Source Succuss Rate (%)		
	Baseline 1 [61]	Baseline 2 [60]	SAGE
Baseline 1 [61]	13.2	9.3	14.4
Baseline 2 [60]	13.5	16.2	13.1
SAGE (Ours)	**39.1**	**24.7**	**46.0**

Table 5. **Cross-Evaluation.** SAGE policies generalize better on out-of-distribution scenes. SAGE even achieves higher success rates on baseline-generated scenes.

final success rate of the policy trained on SAGE-generated data, even with the same number of demonstrations and scenes. To further demonstrate the generalizability of our trained policy, we test on baseline-generated held-out scenes as in Fig. 7. As the results in Tab. 5, despite a distribution shift favoring baselines, our trained policies achieve higher success rates, indicating robust generalization. This comparison highlights that the unique design of SAGE, including agentic orchestration, physics validation, and text-to-3D synthesis, is critical for generating effective training data. Our method not only converges significantly faster than the baselines but also approaches the performance of the privileged agent, demonstrating that high-quality, simulation-validated scenes can efficiently drive policy learning. These results suggest a promising path toward scalable, simulation-driven learning for embodied AI.

5. Conclusion

We present SAGE, an agentic framework that turns open-vocabulary text prompts into *simulation-ready* indoor environments by orchestrating layout and asset generators with visual and physics critics via MCP. The same pipeline scales them for embodied learning through multi-level scene augmentation and automatic action synthesis. On two representative tasks, *i.e.*, Pick-and-Place and Mobile Manipulation, policies trained purely on our generated data show clear scaling with scene diversity and demonstration count, improving generalization to unseen objects and layouts.

Limitations and future work. Our current scope emphasizes indoor scenes and rigid-body physics, and extending to outdoor settings as well as articulated and deformable objects is promising. Action generation presently targets flexible compositions of pick, place, and navigation. Incorporating additional tasks is a natural next step. Beyond imitation, coupling the generator with online RL and real-robot closed-loop validation could further boost performance.

References

[1] Genesis Authors. Genesis: A generative and universal physics engine for robotics and beyond, 2024. 3

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 1

[3] Ata Çelen, Guo Han, Konrad Schindler, Luc Van Gool, Iro Armeni, Anton Obukhov, and Xi Wang. I-design: Personalized llm interior designer. In *European Conference on Computer Vision*, pages 217–234. Springer, 2024. 2

[4] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. 2, 5

[5] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016. 3

[6] Dawson-Haggerty et al. trimesh. 6

[7] 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

[8] Epic Games. Unreal engine. 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. *Advances in Neural Information Processing Systems*, 36:18225–18250, 2023. 2

[10] Rao Fu, Zehao Wen, Zichen Liu, and Srinath Sridhar. Anyhome: Open-vocabulary generation of structured and textured 3d homes. In *European Conference on Computer Vision*, pages 52–70. Springer, 2024. 2, 3

[11] 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. 3

[12] Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 2025. 3

[13] Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Charlie Tianyue Cheng, Haozhe Lou, Peihao Li, Yen-Jen Wang, et al. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning. *arXiv preprint arXiv:2504.18904*, 2025. 3

[14] Zeqi Gu, Yin Cui, Zhaoshuo Li, Fangyin Wei, Yunhao Ge, Jinwei Gu, Ming-Yu Liu, Abe Davis, and Yifan Ding. Artiscene: Language-driven artistic 3d scene generation through image intermediary. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2891–2901, 2025. 2, 3

[15] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7909–7920, 2023. 2

[16] Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 1051–1068, 2023. 3

[17] Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos, 2025. 2

[18] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 3

[19] Steven LaValle. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*, 1998. 5

[20] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021. 3

[21] 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*, pages 80–93. PMLR, 2023. 3

[22] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024. 3

[23] Tsung-Yi Lin, Chen-Hsuan Lin, Yin Cui, Yunhao Ge, Seungjun Nah, Arun Mallya, Zekun Hao, Yifan Ding, Hanzi Mao, Zhaoshuo Li, et al. Genusd: 3d scene generation made easy. In *ACM SIGGRAPH 2024 Real-Time Live!*, pages 1–2. 2024. 2, 3

[24] Lu Ling, Chen-Hsuan Lin, Tsung-Yi Lin, Yifan Ding, Yu Zeng, Yichen Sheng, Yunhao Ge, Ming-Yu Liu, Aniket Bera, and Zhaoshuo Li. Scenethesis: A language and vision agentic framework for 3d scene generation, 2025. 2, 3

[25] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023. 3

[26] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021. 7

[27] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. 3

[28] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit - A Unified Simulation Framework for Interactive Robot Learning Environments. *IEEE Robotics and Automation Letters*, 8 (6), 2023. 2, 7

[29] Model Context Protocol (MCP). Model Context Protocol: Open protocol that standardizes how applications provide context to llms. https://modelcontextprotocol.io/introduction, 2025. Accessed: 2025-11-11. 2, 4

[30] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024. 3

[31] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots, 2024. 7

[32] NVIDIA Corporation. Nvidia isaac sim. https://github.com/isaac-sim/IsaacSim, 2025. Version 5.0.0. 2, 3, 4, 6

[33] OpenAI. Openai api. https://platform.openai.com/docs/api-reference, 2024. Accessed: 2025-11-08. 6

[34] OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. 5

[35] 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, 3

[36] 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. 3

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1

[38] 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 *CVPR*, 2024. 2

[39] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, et al. igibson 1.0: A simulation environment for interactive tasks in large realistic scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7520–7527. IEEE, 2021. 1

[40] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023. 3

[41] F an-Yun Sun, Weiyu Liu, Siyi Gu, Dylan Lim, Goutam Bhat, Federico Tombari, Manling Li, Nick Haber, and Jiajun Wu. Layoutvlm: Differentiable optimization of 3d layout via vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29469–29478, 2025. 2

[42] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. curobo: Parallelized collision-free minimum-jerk robot motion generation, 2023. 2, 5

[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. 3

[44] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20507–20518, 2024. 2, 3

[45] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024. 3

[46] Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 5

[47] 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*, pages 5026–5033. IEEE, 2012. 3

[48] Unity Technologies. Unity, 2023. Game development platform. 3

[49] Giuseppe Vecchio, Renato Sortino, Simone Palazzo, and Concetto Spampinato. Matfuse: Controllable material generation with diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 4429–4438. IEEE, 2024. 4

[50] Boyuan Wang, Xinpan Meng, Xiaofeng Wang, Zheng Zhu, Angen Ye, Yang Wang, Zhiqin Yang, Chaojun Ni, Guan Huang, and Xingang Wang. Embodiedreamer: Advancing real2sim2real transfer for policy training via embodied world modeling, 2025. 1

[51] 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. 3

[52] Yian Wang, Xiaowen Qiu, Jiageng Liu, Zhehuan Chen, Jiting Cai, Yufei Wang, Tsun-Hsuan Wang, Zhou Xian, and Chuang Gan. Architect: Generating vivid and interactive 3d scenes with hierarchical 2d inpainting. *Advances in Neural Information Processing Systems*, 37:67575–67603, 2024. 2

[53] 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. *Transactions on Machine Learning Research*. 3

[54] Hongchi Xia, Zhi-Hao Lin, Wei-Chiu Ma, and Shenlong Wang. Video2game: Real-time, interactive, realistic and browser-compatible environment from a single video, 2024. 2

[55] Hongchi Xia, Chih-Hao Lin, Hao-Yu Hsu, Quentin Leboutet, Katelyn Gao, Michael Paulitsch, Benjamin Ummenhofer, and Shenlong Wang. Holoscene: Simulation-ready interactive 3d worlds from a single video, 2025.

[56] Hongchi Xia, Entong Su, Marius Memmel, Arhan Jain, Raymond Yu, Numfor Mbiziwo-Tiapo, Ali Farhadi, Abhishek Gupta, Shenlong Wang, and Wei-Chiu Ma. Drawer: Digital reconstruction and articulation with environment realism, 2025. 2

[57] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020. 3

[58] 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, 2025. 4, 5

[59] Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. Physcene: Physically interactable 3d scene synthesis for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16262–16272, 2024. 2

[60] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. Holodeck: Language guided generation of 3d embodied ai environments, 2024. 2, 3, 6, 7, 8

[61] Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent, 2025. 2, 3, 6, 7, 8

[62] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *ArXiv*, abs/2210.03629, 2022. 3

[63] Wentao Yuan, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. M2t2: Multi-task masked transformer for object-centric pick and place, 2023. 2, 5

[64] Guangyao Zhai, Evin Pınar Örnek, Shun-Cheng Wu, Yan Di, Federico Tombari, Nassir Navab, and Benjamin Busam. Commonscenes: Generating commonsense 3d indoor scenes with scene graph diffusion. In *Advances in Neural Information Processing Systems*, 2023. 2, 3

[65] Guangyao Zhai, Evin Pınar Örnek, Dave Zhenyu Chen, Ruotong Liao, Yan Di, Nassir Navab, Federico Tombari, and Benjamin Busam. Echoscene: Indoor scene generation via information echo over scene graph diffusion. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 3

[66] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning, 2025. 7