LUMINOUS: Indoor Scene Generation for Embodied AI Challenges

Yizhou Zhao[1*] **Kaixiang Lin**[2] **Zhiwei Jia**[3] **Qiaozi Gao**[2]

Govind Thattai[2] **Jesse Thomason**[2,4] **Gaurav S. Sukhatme**[2,4]

[1]University of California, Los Angeles; [2]Amazon Alexa AI;

[3]University of California, San Diego; [4]University of Southern California

Abstract

Learning-based methods for training embodied agents typically require a large number of high-quality scenes that contain realistic layouts and support meaningful interactions. However, current simulators for Embodied AI (EAI) challenges only provide simulated indoor scenes with a limited number of layouts. This paper presents LUMINOUS, the first research framework that employs state-of-the-art indoor scene synthesis algorithms to generate large-scale simulated scenes for Embodied AI challenges. Further, we automatically and quantitatively evaluate the quality of generated indoor scenes via their ability to support complex household tasks. LUMINOUS incorporates a novel scene generation algorithm (Constrained Stochastic Scene Generation (CSSG)), which achieves competitive performance with human-designed scenes. Within LUMINOUS, the EAI task executor, task instruction generation module, and video rendering toolkit can collectively generate a massive multimodal dataset of new scenes for the training and evaluation of Embodied AI agents. Extensive experimental results demonstrate the effectiveness of the data generated by LUMINOUS, enabling the comprehensive assessment of embodied agents on generalization and robustness. The full codebase and documentation of LUMINOUS is available at: `https://github.com/amazon-research/indoor-scene-generation-eai/`.

1 Introduction

Embodied artificial intelligence (EAI) has attracted significant attention, both in advanced deep learning models and algorithms [1, 2, 3, 4] and the rapid development of simulated platforms [5, 6, 7, 8, 9]. Many open challenges [10, 11, 12, 13] have been proposed to facilitate EAI research. A critical bottleneck in existing simulated platforms [10, 12, 8, 5, 14] is the limited number of indoor scenes that support vision-and-language navigation, object interaction, and complex household tasks. This limitation makes it difficult to verify whether state-of-the-art methods generalize well to unseen scenarios or whether they are specialized to a small number of room structures. Low cost, automatic creation of large numbers of high-quality simulated environments is essential to resolve this question.

Here, we leverage advances in indoor scene synthesis to achieve the large-scale automatic creation of simulated environments. Indoor scene synthesis has been a long-standing challenge for both computer

[*]University of California, Los Angeles. Correspondence to: Yizhou Zhao <yizhouzhao@g.ucla.edu> or Kaixiang Lin <kaixianglin.cs@gmail.com>

1st CtrlGen: Controllable Generative Modeling in Language and Vision Workshop at NeurIPS 2021.

Figure 1: **Generated Indoor scenes.** LUMINOUS scenes are evaluated quantitatively via EAI task success rates and qualitatively via human judgements.

graphics and machine learning communities resulting in considerable recent progress [15, 16, 17, 18, 19, 20, 21, 22, 23]. To effectively utilize indoor scene synthesis for EAI, three key challenges remain. First, for synthesized scenes to be useful in EAI, they must directly support household tasks requiring object pick and place, state changes, and articulation. Second, the generated scenes with randomized layouts must be *natural*—layouts that "make sense" according to human judgement—and *functional*—layouts that match human use given the room type, such as *Bedroom* or *Living Room*. Finally, any scene generation method must provide efficient access to massive, multimodal embodied agent trajectory data, including low-level action sequences for task completion, egocentric image frames during action execution, and language instructions.

We present LUMINOUS, a scalable, indoor scene generation framework to facilitate EAI tasks such as vision-and-language navigation and language-guided task completion (Figure 1). We introduce the Challenge Definition Format (CDF), which provides a user-friendly task specification of the required objects, their relative spatial relationships, and high-level descriptions of downstream EAI tasks to facilitate. We introduce Constrained Stochastic Scene Generation (CSSG) to generate an arbitrary number of indoor scenes from the CDF specification. LUMINOUS produces scenes that are well-aligned with human common sense and satisfy the CDF conditions, thereby ensuring that the generated scenes are readily applicable to EAI tasks. In addition, we develop a task solver to plan sequences of low-level actions for corresponding task completion. We also implement a task instruction generation module to annotate trajectories with language instructions. LUMINOUS generates large-scale multimodal trajectories for the training and evaluation of embodied agents.

LUMINOUS also contributes to indoor scene synthesis. Generally, scene generation lacks ground truth for quantitative evaluation. Metrics like bounding box and angle prediction [20] and synthetic classification [19] are not always correlated with the quality of a generated scene. By connecting indoor scene synthesis to EAI, we propose measuring planner-based task success rate as an automatic evaluation metric of the synthesized scene quality. Besides CSSG, LUMINOUS is compatible with state-of-the-art learning-based indoor scene synthesis algorithms [24, 20]. We demonstrate that CSSG with LUMINOUS qualitatively outperforms other learning-based synthesis methods (Section 4.1).

The main contributions of our work are threefold. First, we introduce a framework (LUMINOUS) which serves as a standard and unified benchmark for indoor scene synthesis algorithms. Second, LUMINOUS generates a large number of randomized scenes that achieve competitive quality compared to human-designed scenes in AI2Thor [6]. Third, the rendered scenes, along with the multimodal trajectories, directly support typical EAI task completion to facilitate generalization research. Extensive evaluation on ALFRED [10], a language-guided task completion challenge, demonstrate the effectiveness and scalability of LUMINOUS. Further, our evaluation with LUMINOUS scenes suggests that existing, state of the art models for ALFRED may overfit to the hand-created scenes in AI2Thor.

2 Related Work

LUMINOUS builds on and extends research in indoor scene synthesis, simulation environments in EAI, and language-guided task completion.

Indoor Scene Synthesis. In computer graphics, extensive research exists in 3D indoor scene synthesis. Early work either used explicit rule-based constraints [25] or incorporated stochastic priors into the generative procedure [15, 16, 17, 18]. Recent advances [19, 20, 22] utilize deep neural networks to extract patterns from large-scale datasets [26]. While these data-driven approaches significantly

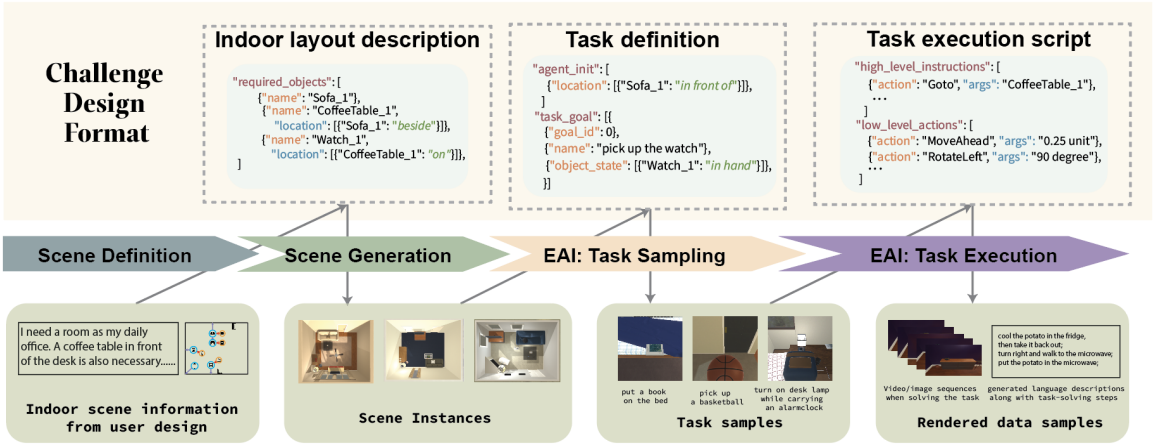


Figure 2: **The Luminous Framework.** Scene definitions constrain generated scenes, which are pragmatically evaluated via household task sampling and execution to ensure generated scene quality.

enhance the automation of the scene generation process, the resulting synthesized scenes are still relatively simple in terms of object quantity and inter-object spatial relationships. Many works generate scenes based on the natural representation of the scene graph [21, 19, 20]. Other lines of research condition on the image [24, 27] or text [28, 29] representation of indoor scenes. The discrepancies in the input representation of scene generation models and the diverse sources of data make it difficult to compare and contrast the performance of different methods. To facilitate research in learning-based approaches, LUMINOUS is designed to support end-to-end scene generation evaluation and a unified rendering tool to accommodate the outputs of various approaches simultaneously.

Embodied AI Simulators. In the past few years, researchers have developed many simulation environments [6, 7, 13, 5, 9] to serve as training and evaluation platforms for embodied agents. These simulation environments propel research progress in a wide range of embodied tasks, including vision-and-language task completion [10, 30], rearrangement [12, 7], navigation [9, 13], manipulation [31, 32] and human-robot collaboration [5]. Recently, AllenAct [33] integrates a set of embodied environments (such as iThor, RoboThor, Habitat [9], etc.), tasks, and algorithms thereby facilitating the evaluation of the same model or algorithm across multiple EAI platforms. Many EAI platforms are designed with sophisticated indoor scenes to perform embodied tasks. Platforms such as iGibson [13], AI2Thor [6] can randomize materials, color, and small objects in the scene, while the basic room layouts remain unchanged. To facilitate more robust and thorough evaluation of embodied agents, LUMINOUS automatically generates indoor scenes with randomized layouts at a large scale that readily support vision-and-language navigation and high-level object interactions. We summarized the properties of LUMINOUS and most popular EAI simulation platforms in Table 9.

Language-Guided Task Completion. Among existing EAI challenges, we use ALFRED [10] as our downstream exemplar task to evaluate the scene generation quality of LUMINOUS. ALFRED enables agents to follow natural language descriptions to complete complex household tasks. ALFRED tasks involve resolving vision-and-language grounding, affordance-aware navigation, and high-level object interactions. Roughly speaking, there are two categories of approaches to tackling ALFRED. Initial approaches learned end-to-end models that mapped language instructions into low-level actions directly [30, 3, 34]. Subsequently, hierarchical approaches [4, 35] were proposed that enabled better generalization and interpretation. However, those approaches are only tested in four indoor scenes unseen during training time. Towards a more convincing evaluation, LUMINOUS generates an order of magnitude larger number of scenes for better assessment of generalization and robustness.

3 LUMINOUS

LUMINOUS bridges the fields of indoor scene generation and EAI task completion. A well-designed indoor scene needs to support different daily tasks. Accordingly, LUMINOUS generates an unlimited

number of randomized layouts for EAI training and evaluation, while using the task success rate of an oracle planner as an automatic metric to evaluate the quality of the generated scenes.

3.1 Framework Overview

The scene generation pipeline of LUMINOUS consists of four stages, as shown in Figure 2. First, in the SCENE DEFINITION stage, users specify the required objects and, optionally, objects' relative spatial relationships. In the SCENE GENERATION stage, we propose a Constrained Stochastic Scene Generation (CSSG) algorithm to synthesize scenes whose layouts are randomized while satisfying user requirements and incorporating common sense knowledge to encourage scenes to be natural and functional. Next, the TASK SAMPLING stage programmatically samples household tasks that are executable in the current scene. Finally, the TASK EXECUTION stage plans a sequence of low-level actions for the agent to execute to complete the task, and generates a series of natural language instructions to describe the agent's behavior.

3.2 Challenge Definition Format

We introduce the Challenge Definition Format (CDF) to concurrently support the description of indoor layouts and the execution of household tasks (Figure 2). Learning-based indoor scene synthesis approaches are restrictive for generating EAI simulated environments [36]. For example, these predict absolute locations for meshes, voxels, or point clouds for objects. By contrast, humans naturally understand the layout of an indoor scene in terms of the relative relationships among objects, such as a coffee cup on a table, a bed against a wall, and a chair in front of a desk. Recent scene synthesis algorithms such as Planit [19] and 3D-SLN [20] have demonstrated the effectiveness of using a directed graph to store the relative positions of furniture. Based on this insight, we argue that relative object relationships are more important than the absolute locations of objects for understanding the functional and intrinsic utility of the room. Anecdotally, we feel specifying scene layouts through relative object relationships is more flexible and user-friendly than absolute coordinates. In the indoor layout description section of the CDF, we define the required objects that must exist in the scene, including furniture, household items, and decorations, along with the relationship among those objects, for example that a book is on a table. Figure 2 shows an example of the indoor layout description. Each entry holds the name, type, or class of an item and may optionally have its spatial relation relative to another object. In addition, similar to 3D-SLN [20], attributes such as color, material, and size can also be attached to an entry to further describe the object.

The CDF also contains of a task definition section and a task execution script. Instead of being specified by users, these sections can be automatically generated via the task sampling stage and the task execution stage. The task definition section specifies the task to be completed within the scene. The execution script lists out the action sequences for completing the task. Within the task definition section, inspired by Planning Domain Definition Language (PDDL) [37, 10], the CDF defines the initial state of the scene, comprising the position of the agent and the states of objects, and the conditions for task completion, for example that a desk lamp is toggled on. Figure 2 shows an example of an EAI task definition. The CDF can contain the execution script for the task in the form of human-understandable (high-level) instructions and atomic (low-level) actions.

3.3 Constrained Stochastic Scene Generation

To stochastically generate high-quality indoor scenes satisfying the layout constraints defined in the CDF, we propose a novel method: Constrained Stochastic Scene Generation (CSSG). Inspired by the energy-based indoor scene synthesis method [18], CSSG generates scenes in a hierarchical manner, which enables great flexibility to enforce constraints and to incorporate prior knowledge. First, CSSG samples the room structure, such as walls, floors, and windows, from a set of pre-defined candidates. Next, CSSG samples types, positions, and rotations of large furniture defined in the CDF. During sampling, unlike human-centric indoor scene synthesis which learns the distribution of furniture from data, CSSG generates the distribution of the position and orientation of furniture according to *relationships* among furniture and room structure. Next, CSSG places objects in or on specific furniture, for example placing a coffee machine on a dining table. Finally, CSSG optionally generates decorations such as wall paintings and carpets.

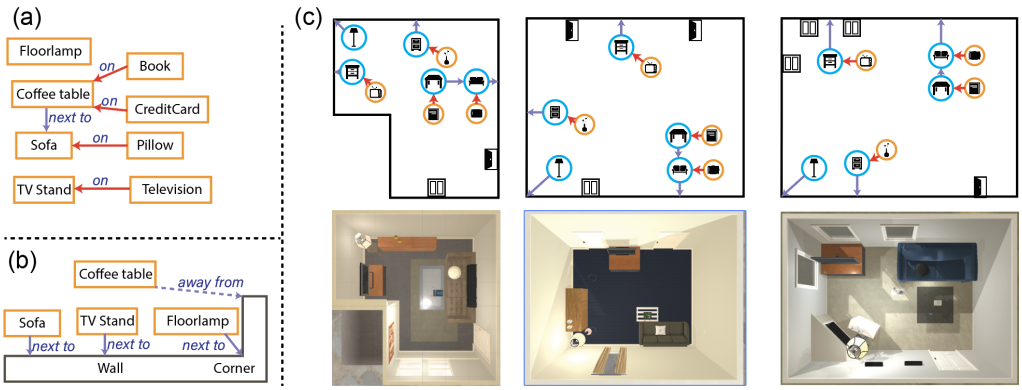


Figure 3: **CSSG illustration.** (a) explicit relationships defined in the CDF; (b) implicit relationships added by LUMINOUS; (c) sampled scenes satisfying relationships defined in (a) and (b) with different room structures.

Apart from the relationships defined explicitly in the CDF file, CSSG also integrates implicit relationships based on common sense. For example, if the CDF specifies "a bed is beside a reading desk", CSSG adds an implicit rule "the bed is against the wall" when sampling the position of the bed. When multiple relationships influence the position of an object, we use a set of predefined weights for different types of relationships. Experimental results (Section 4.1) show that the *rule-based* CSSG with predefined weights can reasonably balance human prior knowledge with the constraints specified in the CDF thus generating meaningful and functional indoor scenes. Therefore, LUMINOUS adopts CSSG as the default scene generation algorithm for EAI evaluation. We refer readers to Section A.2 in the Appendix for details on implicit relationships, types of relationships, and predefined weights. Figure 3 illustrates the scene generation pipeline of CSSG and shows several sample scenes generated by CSSG, with more in Appendix Section C.

3.4 Automatic EAI Task Sampling and Task Execution

Another challenge of using traditional indoor scene synthesis for EAI tasks is the lack of logic inherent to object interaction, state changes, and agent actions. It is unclear how to enable complex interaction capabilities within the framework of prior scene generation algorithms. To enable consideration of object interaction constraints, LUMINOUS is implemented on top of the interactive 3D platform AI2Thor [6], which possesses 102 interactive object types, more than 2000 3D meshes, and most importantly: physical interaction mechanisms. We seamlessly connect the high-quality indoor scenes generated by CSSG and the sophisticated physical interaction logic provided by AI2Thor. LUMINOUS can thus directly support many complicated EAI challenges, including but not limited to ALFRED [10], Rearrangement [38], and RoboTHOR [39].

Given generated scenes, LUMINOUS can utilize the planner proposed in ALFRED [10] to sample solutions to simulation tasks. Additionally, given the tasks, LUMINOUS can resolve and generate appropriate scenes to support those EAI tasks. For details on task generation with ALFRED, see Section 3.6. Note that the task generation in LUMINOUS does not rely on ALFRED challenges. With the CDF used in LUMINOUS, we can easily sample an arbitrary number of simple tasks.

The task execution stage in LUMINOUS decomposes a household task into *navigation* and *interaction* tasks. *Navigation* requires the agent to find an optimal route from one place to another while avoiding collisions, which is achieved by a planner inside of LUMINOUS. *Interaction* often requires the agent to trigger the state change of certain object. For example, "taking a book on the coffee table" can be decomposed into the navigation part "go to a coffee table" and the interaction part "pick up the book". LUMINOUS applies Dijkstra's algorithm to get the shortest path for navigation, and AI2Thor's interaction mechanism to perform the agent-object interaction.

LUMINOUS provides two methods to generate natural language descriptions for household tasks involving navigation and object interactions. The first method relies on a rule-based language template to generate language instructions for different tasks (See Appendix Section B). The second method

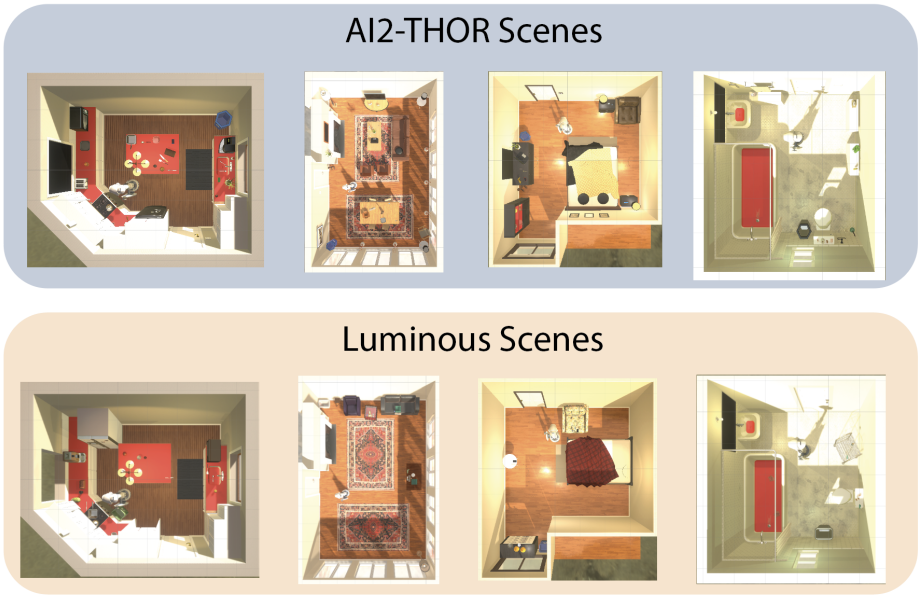


Figure 4: **Sample AI2Thor and LUMINOUS scenes for EAI challenges.** For kitchens and bathrooms, LUMINOUS keeps more parts of the room structures. See Appendix C for more details.

uses the *Speaker* model proposed in Episodic Transformer [34] that maps the low-level actions and corresponding egocentric images into generated language task instructions.

3.5 Accommodating Learning-based Indoor Scene Synthesis

Apart from the energy-based approach (CSSG), LUMINOUS incorporates two learning-based indoor scene synthesis methods, 3D-SLN [20] and Deep-synth [24], by training indoor-scene generators from the 3D-FRONT dataset [40]. An obstacle that hinders the application of most learning-based methods to EAI tasks are object model discrepancies between the indoor-scene dataset and EAI simulators. LUMINOUS accommodates indoor scenes generated by 3D-SLN and Deep-synth by matching model names, furniture sizes, and room shapes between 3D-FRONT and AI2Thor, thereby providing a unified interface for learning-based approaches to train on the 3D-Front dataset and generated scenes with AI2Thor assets. For details, see Appendix Section A.1.

3.6 LUMINOUS for ALFRED: A Comprehensive Example

We apply LUMINOUS to ALFRED, a benchmark for learning a mapping from natural language instructions and egocentric vision to sequences of actions for household tasks. The goal is to automatically generate additional data by LUMINOUS that shares exactly the same format as ALFRED training and evaluation data.

Given a trajectory T_i from the ALFRED training dataset, we employ a task parser to deduce objects and their relationships and save the scene conditions into the indoor-scene description part I_i of CDF. Since each training scene in ALFRED supports dozens of trajectories $\{T_i\}_{i=1,2,...}$, there may be some conflicting parts in their scene description $\{I_i\}_{i=1,2,...}$. For example, one task requires $\{Apple\_1\}$ to be on the countertop; another says $\{Apple\_1\}$ should be in the fridge. We propose a $merge$ operator $merge(I_1, I_2, ...) \rightarrow \hat{I}$, where $\hat{I}$ denotes the merged links in indoor-scene description file, that tries to maximize common parts in the scene descriptions to tackle this problem. We use this merge operation for sampling indoor scene layouts S by CSSG. Since ALFRED does not change the positions of large pieces of furniture, such as fridges, sofas, and beds, the $merge$ operator records the requirements for large pieces of furniture and extracts the most common criteria for small objects (e.g., apple, cup, and book). Figure 4 shows the comparison between AI2Thor original scenes and LUMINOUS scenes generated to augment the ALFRED challenge.

	Method	# Scenes, # Ratings	Functionality (1-5)	p-value vs. LUMINOUS	Naturalness (1-5)	p-value vs. LUMINOUS
Generated	Deep Priors	50, 150	2.40 ± 1.40	**$\sim$ .0**	1.78 ± 1.06	**$\sim$ .0**
	3D-SLN	50, 150	2.45 ± 1.43	**$\sim$ .0**	2.03 ± 1.35	**$\sim$ .0**
	LUMINOUS	50, 150	**4.13** ± 1.00		**3.83** ± 1.11	
Human	AI2Thor	30, 90	4.23 ± 0.97	.416	3.68 ± 1.07	.308

Table 1: **Human subjects' ratings of the functionality and naturalness of *Bedroom* scenes.** LUMINOUS is rated statistically significantly better than existing, state-of-the-art generation methods.

After obtaining an indoor scene S, we apply two techniques to sample tasks and trajectories. The first follows the Fast-Forward Planner (FF-Planner) [10] and samples tasks and trajectories by sequentially setting initial conditions, sampling task goals, and executing trajectories. The second follows the original task design D_i and directly applies the *task execution* component to generate the trajectory T_i'. Locations of small objects defined by I_i must be resampled for each task before execution.

The FF-Planner is slower at sampling tasks because it experiences trial and error in different sampling stages. We compare the efficiency of this method between sampling from AI2Thor original scenes and from LUMINOUS-generated scenes in Section 4.1. The sampling efficiency indicates the quality of the indoor scene. The second method samples trajectories much faster since it directly applies the task design D_i from original ALFRED training data which can be quickly solved by the TASK EXECUTION stage in LUMINOUS. We apply this method to generate a large number of scenes for the evaluation performance of different models in Section 4.2.

4 Experiments

We evaluate LUMINOUS both quantitatively and qualitatively. Our experiments focus on answering the following questions: 1) LUMINOUS *for indoor scene synthesis*: Does LUMINOUS generate high-quality scenes that are aligned with human common sense? 2) LUMINOUS *for EAI*: How well do the generated scenes support downstream EAI tasks? 3) *EAI task evaluation with* LUMINOUS: Can LUMINOUS generate indoor scenes that serve as reliable evaluation environments for EAI tasks? In addition, we discuss the insights obtained from the evaluation of state-of-the-art language-guided task completion models with larger set of unseen environments generated via LUMINOUS.

4.1 The Quality of LUMINOUS-generated Scenes

To answer the first two questions on evaluating the quality of LUMINOUS generated scenes from the perspective of both human common sense and the capability of supporting EAI tasks, we conduct user studies and oracle task success rate. We further demonstrate the great variety of tasks supported on scenes generated by LUMINOUS.

User Studies: Following the evaluation protocol proposed in [18], we conducted user studies on Amazon Mechanical Turk comparing the quality of *Bedroom* scenes generated by LUMINOUS with two state-of-the-art learning-based approaches: Deep Priors [23] and 3D-SLN [20]. Generated scenes are shown to users without any post-processing such as removing bad samples. Additionally, we compared LUMINOUS scenes against human-designed scenes in AI2Thor [6]. Users were asked to evaluate scene quality, with scenes given as top-view images (Figure 4), based on two criteria: functionality and naturalness. Functionality describes how the room layout satisfies a human's needs for daily life. Naturalness indicates whether the room layout is realistic. Scales of responses range from 1 to 5, with 5 indicating perfect functionality or naturalness. For every scene, we collect three ratings per metric. The mean ratings and standard deviations are summarized in Table 1. LUMINOUS achieves competitive performance with the human-designed scenes in AI2Thor [6]. We ran six Welch's unpaired, two-tailed t-tests to compare LUMINOUS scores with those of AI2Thor and the learning-based approaches on both metrics. After a Bonferroni multiple-comparison correction, we find that LUMINOUS scenes are rated statistically significantly more functional and natural than scenes from both Deep Priors and 3D-SLN, the learning-based approaches, and not significantly differently from human-designed AI2Thor scenes.

Task Success Rate: Our proposed framework for indoor scene generation aims to promote better training and evaluation of the Embodied AI tasks. We show that, powered by the Constrained

	Task Success Rate	
	AI2Thor (Human)	LUMINOUS (Generated)
Pick & Place	.33	.13 (Δ-.20)
Pick Two & Place	.10	.06 (Δ-.04)
Examine in Light	.55	.59 (Δ .04)
Clean & Place	.18	.17 (Δ-.01)
Heat & Place	.19	.09 (Δ-.10)
Cool & Place	.07	.07 (Δ .00)
Stack & Place	.05	.09 (Δ .04)
Overall	.21	.17 (Δ-.04)

	Subgoal Success Rate	
	AI2Thor (Human)	LUMINOUS (Generated)
Heat Object	.19	.09 (Δ-.10)
Cool Object	.07	.07 (Δ .00)
Clean Object	.18	.17 (Δ-.01)
Slice Object	.11	.09 (Δ-.02)
Put Object	.15	.10 (Δ-.05)
Toggle Object	.55	.59 (Δ .04)
Pickup Object	.22	.18 (Δ-.04)
Goto Location	.21	.17 (Δ-.04)

Table 2: **Left: Task Success Rate.** For most task types, the loss in success rate between AI2Thor human-created scenes and LUMINOUS generated scenes is less than 5%, and for some tasks success rate improves. **Right: Subgoal Success Rate.** Multiple subgoals are carried out for each task. The loss in success rate in LUMINOUS generated scenes is usually less than 5%, and sometimes improves.

			Trajectories per Task Type							
Split	**Scene**		Pick	Pick Two	Examine	Clean	Heat	Cool	Stack	**Overall**
Seen	AI2Thor	(S)	46	33	29	27	34	38	34	**251**
	LUMINOUS	(S+)	226	167	236	210	163	202	201	**1405**
Unseen	AI2Thor	(U)	30	24	54	36	42	36	33	**255**
	LUMINOUS	(U+)	27	18	178	56	21	56	79	**435**

Table 3: **Validation Trajectory Counts by Task Type.** ALFRED trajectories were sampled from both human-created AI2Thor scenes and generated LUMINOUS scenes to evaluate EAI agents.

Stochastic Scene Generation strategy, LUMINOUS procedurally generates indoor scenes that can produce high-quality trajectories for downstream navigation and object manipulation tasks in a comparable level of efficiency even to the manually-designed scenes provided by the ALFRED [10] dataset. We adopt the same task sampling strategy as in the ALFRED dataset, which roughly samples 200 tasks for each of the 7 task types (Pick & Place, Stack & Place, Examine in Light, etc.) The tasks designed in the ALFRED dataset involve long-horizon navigation and object manipulations in indoor scenes and are very challenging such that even those sampled in the hand-designed scenes fail to be solved most of the time by a carefully-tuned Planning Domain Definition Language (PDDL) rule-based [41] motion planner. Here we present the task success rate for a given set of scenes, defined as the percentage of tasks randomly sampled in the scenes that can be successfully solved by a rule-based, oracle planner. To make a fair comparison, we use the same sampling strategy and motion planner provided by the ALFRED dataset. As similar to the training fold in ALFRED, we construct 108 scenes by using LUMINOUS (26 scenes for each of the 4 room types). We compare the task success rate of these scenes with the rate of the manually designed scenes from AI2Thor [6]. Our scene generation algorithm is automatic, and does not leverage knowledge of the motion planner in ALFRED that is tailored towards AI2Thor scenes.

Subgoal Statistics: Scenes generated by LUMINOUS support a large variety of (sub-)tasks introduced as "subgoals" in the ALFRED dataset. Each task in ALFRED consists of several subgoals ranging from navigation to object manipulations such as "SliceObject" and "ToggleObject". In total there are 8 types of subgoals and we calculate the statistics of these subgoals in tasks sampled from scenes as described above. See Table 2 (Right) for the comparison between LUMINOUS and AI2Thor. This subgoal level evaluation further reveals appealing properties of LUMINOUS. For example, LUMINOUS achieves 17% task success rate in the GotoLocation subgoal, which indicates the generated scene has a comparable connectivity with human-created scenes in AI2Thor and the robot can move freely across a large portion of scene using a simple planner that does not account for held-object collisions.

4.2 LUMINOUS as an EAI Evaluation Platform

We use LUMINOUS to provide two different settings to evaluate state-of-the-art inference models for the ALFRED challenge. All simulated scenes, trajectories, and task instructions are generated by LUMINOUS. In the first setting, we use the room structures (the shape of floor, wall, and ceiling) in the *unseen* validation set of ALFRED, and then apply LUMINOUS to randomize the scene layouts and sample the tasks and trajectories under the same room structures. For each of the four rooms' structures in the validation *unseen* set, we sample four room layouts and dozens of tasks. For each task,

	ALFRED Inference Model											
	MOCA				ET				HiTUT			
Task	S	S+	U	U+	S	S+	U	U+	S	S+	U	U+
Pick	.295	.131	.005	.429	.500	.227	.040	.381	.359	.314	.260	.259
Cool	.261	.000	.070	.000	.532	.035	.010	.018	.190	.035	.046	.034
Stack	.052	.000	.018	.000	.296	.025	.028	.000	.122	.065	.073	.038
Heat	.158	.000	.027	.000	.458	.000	.074	.000	.140	.061	.119	.000
Clean	.223	.000	.024	.000	.482	.129	.170	.109	.500	.229	.212	.232
Examine	.202	.000	.132	.000	.426	.072	.070	.034	.266	.173	.081	.067
Pick Two	.112	.011	.011	.000	.419	.034	.051	.000	.177	.096	.124	.111
Average	.186	.022	.038	.021	.448	.078	.066	.048	.252	.147	.124	.090

Table 4: **Success rate on ALFRED tasks across validation splits.** S: ALFRED *seen*; U: ALFRED *unseen*; U+ *Unseen Plus via* LUMINOUS; S+ *Seen Plus via* LUMINOUS. Note that all ALFRED models, in both *seen*- and *unseen*-based layouts, suffer loss of performance when generalizing to generated LUMINOUS scenes for nearly every task.

we sample one trajectory to solve the task. In total, we generate 16 indoor scenes and 435 trajectories. In the second setting, we randomly take 10 room structures in the *training* set of ALFRED for each room type (*Kitchen*, *Living Room*, *Bedroom*, and *Bathroom*). Then, with the 40 room structures, we randomize one layout and dozens of tasks for each. The second setting produces 1405 trajectories for evaluating EAI models, which is an order of magnitude larger than ALFRED *unseen* in terms of both task numbers and scene numbers. Table 3 summarizes the number of trajectories for each task type in ALFRED validation *seen*, *unseen*, and the two evaluation settings empowered by LUMINOUS.

With the aforementioned four test settings, we evaluate three top-ranked models for ALFRED challenge: MOCA [30], Episodic Transformer (ET) [34], and HiTUT [4] on LUMINOUS validation settings. We denote the first validation setting as Unseen Plus (U+) and the second as Seen Plus (S+). For the validation performance of MOCA and HiTUT on ALFRED *seen* and *unseen*, we directly report their performance described in the paper. For the experimental results of ET, we evaluate its performance based on the checkpoints provided by the authors of ET.

In Table 4, we show the overall performance and per-task type's for MOCA, ET, and HiTUT. First, we found that the relative performance of the three models in our setting is generally consistent with ALFRED's overall generalization performance, where HiTUT achieves the best performance among the three models, and ET outperforms MOCA. It indicates that the models that perform well in the ALFRED challenge adapt to our randomized scenarios and tasks. However, comparing the evaluation results in unseen environments (U vs U+), there is a notable drop in generalization performance when we increase the number of test scenes from 4 to 16. This confirms that the current evaluation in ALFRED might not provide "true" generalization evaluation and highlights the significance of LUMINOUS for the embodied AI research. Second, we notice that the performance under S+ is similar to ALFRED *unseen* (U) in terms of large performance drop compared to ALFRED *seen* (S), even though the scenes and tasks generated by LUMINOUS share the same room structure (including walls, windows, doors, etc.) with scenes in ALFRED's training. The randomized layouts from LUMINOUS that produce different locations of objects introduce extra difficulties for the models to accomplish tasks. It is worth noting that the high success rate of Pick tasks is due to LUMINOUS place the object in the edge of receptacles (e.g., table, shelf, sofa, etc.). This provides a broader range of areas for the robot to pick up the objects and thus leads to a much higher success rate than other task types.

5 Conclusion

We introduced LUMINOUS, a framework to *illuminate* general indoor scene generation for EAI challenges. LUMINOUS generates large-scale, high-quality simulated indoor scenes that are competitive with manually designed scenes in terms of naturalness and their ability to support various EAI tasks. Extensive empirical results on language-guided task completion challenges demonstrate the effectiveness of LUMINOUS to serve as a reliable and useful EAI evaluation platform.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[2] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

[3] Alessandro Suglia, Qiaozi Gao, Jesse Thomason, Govind Thattai, and Gaurav Sukhatme. Embodied bert: A transformer model for embodied, language-guided visual task completion. *arXiv preprint arXiv:2108.04927*, 2021.

[4] Yichi Zhang and Joyce Chai. Hierarchical task learning from language instructions with unified transformers and self-monitoring. *arXiv preprint arXiv:2106.03427*, 2021.

[5] 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.

[6] 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 preprint arXiv:1712.05474*, 2017.

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

[8] 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.

[9] 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.

[10] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.

[11] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. *arXiv preprint arXiv:2106.14405*, 2021.

[12] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[13] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Shyamal Buch, Claudia D'Arpino, Sanjana Srivastava, Lyne P Tchapmi, et al. igibson, a simulation environment for interactive tasks in large realisticscenes. *arXiv preprint arXiv:2012.02924*, 2020.

[14] Claudia Yan, Dipendra Misra, Andrew Bennnett, Aaron Walsman, Yonatan Bisk, and Yoav Artzi. Chalet: Cornell house agent learning environment. *arXiv preprint arXiv:1801.07357*, 2018.

[15] Lap Fai Yu, Sai Kit Yeung, Chi Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics (TOG)-Proceedings of ACM SIGGRAPH 2011, v. 30,(4), July 2011, article no. 86*, 30(4), 2011.

[16] Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. Example-based synthesis of 3d object arrangements. *ACM Transactions on Graphics (TOG)*, 31(6):1–11, 2012.

[17] Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. Activity-centric scene synthesis for functional 3d scene modeling. *ACM Transactions on Graphics (TOG)*, 34(6):1–13, 2015.

[18] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5899–5908, 2018.

[19] Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019.

[20] Andrew Luo, Zhoutong Zhang, Jiajun Wu, and Joshua B Tenenbaum. End-to-end optimization of scene layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3754–3763, 2020.

[21] Yang Zhou, Zachary While, and Evangelos Kalogerakis. Scenegraphnet: Neural message passing for 3d indoor scene augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7384–7392, 2019.

[22] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. *arXiv preprint arXiv:2012.09793*, 2020.

[23] Zaiwei Zhang, Zhenpei Yang, Chongyang Ma, Linjie Luo, Alexander Huth, Etienne Vouga, and Qixing Huang. Deep generative modeling for scene synthesis via hybrid representations. *ACM Transactions on Graphics (TOG)*, 39(2):1–21, 2020.

[24] Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[25] Ken Xu, James Stewart, and Eugene Fiume. Constraint-based automatic placement for scene composition. In *Graphics Interface*, volume 2, pages 25–34, 2002.

[26] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1754, 2017.

[27] Daniel Ritchie, Kai Wang, and Yu-an Lin. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6182–6190, 2019.

[28] 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.

[29] Angel Chang, Manolis Savva, and Christopher D Manning. Semantic parsing for text to 3d scene generation. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 17–21, 2014.

[30] Kunal Pratap Singh, Suvaansh Bhambri, Byeonghwi Kim, Roozbeh Mottaghi, and Jonghyun Choi. Moca: A modular object-centric approach for interactive instruction following. *arXiv preprint arXiv:2012.03208*, 2020.

[31] 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.

[32] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[33] Luca Weihs, Jordi Salvador, Klemen Kotar, Unnat Jain, Kuo-Hao Zeng, Roozbeh Mottaghi, and Aniruddha Kembhavi. Allenact: A framework for embodied ai research. *arXiv preprint arXiv:2008.12760*, 2020.

[34] Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic transformer for vision-and-language navigation. *arXiv preprint arXiv:2105.06453*, 2021.

[35] Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. A persistent spatial semantic representation for high-level natural language instruction execution. *arXiv preprint arXiv:2107.05612*, 2021.

[36] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. 3d semantic scene completion: a survey. *arXiv preprint arXiv:2103.07466*, 2021.

[37] Patrik Haslum, Nir Lipovetzky, Daniele Magazzeni, and Christian Muise. An introduction to the planning domain definition language. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(2):1–187, 2019.

[38] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020.

[39] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, Luca Weihs, Mark Yatskar, and Ali Farhadi. Robothor: An open simulation-to-real embodied ai platform. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[40] Huan Fu, Bowen Cai, Lin Gao, Lingxiao Zhang, Cao Li, Zengqi Xun, Chengyue Sun, Yiyun Fei, Yu Zheng, Ying Li, et al. 3d-front: 3d furnished rooms with layouts and semantics. *arXiv preprint arXiv:2011.09127*, 2020.

[41] Constructions Aeronautiques, Adele Howe, Craig Knoblock, ISI Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, David Wilkins SRI, Anthony Barrett, Dave Christianson, et al. Pddl| the planning domain definition language. Technical report, Technical Report, 1998.

A LUMINOUS

A.1 Details on incorporating Learning-based Indoor Scene Synthesis to LUMINOUS

As we shown in Figure 2, the overall structure of LUMINOUS mainly consists of three components. First, we propose a unified representation of indoor scene processing, providing various interfaces for data processing, making the original data in different formats required by different models: e.g. RGB images, bounding boxes with object types, etc. After that, different data formats are used as inputs to different models for training indoor scene generation models. It is worth noting that we unify the model-generated scene formats again, allowing us to use the same scene rendering tools to automatically visualize the scenes. Finally, we provide different testing interfaces to uniformly evaluate the quality of various algorithm-generated scenarios.

Data processing
Since our ultimate task is to provide indoor scenes as experimental environments for Embodied AI, the data we target should provide a full set of information about the indoor scenes: e.g., house structure, furniture models, and object placement information. Luminous selects three data sources for data processing: mesh information from 3D-FRONT [40], and game designs from AI2Thor [6]. In the data processing, we first unify the names of items in different datasets (e.g. *picture = painting*, *bedside cabinet = nightstand*). The full list of unified furniture and object names are attached in the appendix. Then we normalize the coordinated w.r.t. locations and rotations. We also normalize room scales. Finally, according to different formats of the training data for different methods, we generally provides three different data formats: RGB-D images, semantic segmentation, and bounding boxes together with object types and rotations.

Scene Synthesis
Luminous provides some state-of-the-art algorithms for indoor scene synthesis. We chose Python as programming language ,and Pytorch for deep learning. We have carefully referred to the source code of these these methods. However, for the reason such as missing public training dataset, and the compromise we have made for unifying data formats (e.g. *double bed* $\rightarrow$ *bed*), the re-implemented performance in Luminous for those methods may differ from the original one.

A.2 Constrained Stochastic Scene Generation

We consider the problem of indoor scene generation under certain constraints represented by text descriptions [28] or scene graphs [20]. In our baseline, each constraint not only defines the type of an object, but also optionally describes the object's relationship with others in the scene. In detail, a constraint c_i provides the information for placing object i by defining its type o_i (e.g. *bed*), and a set of relationship with others $R_i = \{rel(i, j_k)\}_{k=1,2,...}$, where j_k stands for another object in the scene and $rel(\cdot, \cdot)$ specifies the relationship between two objects (e.g. *bed* ***beside*** *window*).

Given a set of constraints $\{c_i\}_{i=1,2,...}$ and the room structure (the shape of floor, wall and ceiling), an indoor scene is sampled from a sequential process of three layers. The first layer samples pieces of **furniture** that represent the overall function of the room and can be placed directly on the floor, such as *bed, dinning table*, and *refrigerator*. The second layer samples **objects** that are usually supported by another piece furniture such as *book, pen*, and *coffee machine*. Finally, the third layer samples **decorations** in the scene such as *painting* and *carpet*.

In each layer, we empirically defined the priority value $q(i)$ as the order for placing furniture according to object types. For example, we prefer to place *desk* before placing *chair*: $q(desk) > q(chair)$. Besides, we limit the constraints that can be represented by a direct acyclic graph (DAG) and resolve the relationship between objects to ensure that when calculating $rel(i, j_k)$, we have $q(i) > q(j_k)$. For example, if the text description says *a desk is in front of a chair*, it is resolved as *a chair faces a desk*.

When placing each object, we samples the position and rotation of the object by its explicit relationship with others $\{rel(i, j_k))\}_{k=1,2,...}$ defined previously, and implicit relationship with others $\{\widetilde{rel}(i, j_k))\}_{k=1,2,...}$ predefined heuristically from our prior knowledge. For example, humans are in favor of pushing the *bed* up against the *wall* of a *Bedroom* $(bed, (wall, against))$.

Each relationship $rel(i, j_k)$ generates a vector field in space: each position p is characterized by $(s_{p,k}, r_{p,k})$, where $s_{p,k}$ is the score of the point. s_i depends on the distance d_i between p and the target object j_k. Figure 5(a) shows different types of relationship and the scores deduced by the

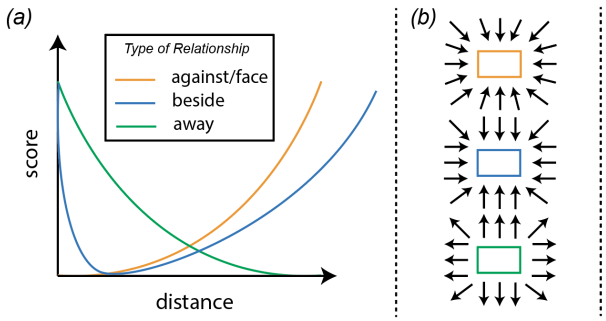


Figure 5: Illustration of how to sample the position of an object according to the type of relationship. (a) Score functions for different types of relationship, depended on the distance between the sampling position p and the target object j_k. (b) Direction vectors suggesting the rotation $r_{p,k}$ of the object being placed on the position.

relative distance. $r_{p,k}$, suggesting the relative rotation of placing the object, depends on the direction vector from p to its target j_k the type of relationship. Combining $s_{p,k}$ with parameter $w_{type(rel(i,j_k))}$ related only the type of relationship, we sample the position to place object i according to weighed sum of scores among all relationship, and the rotation of the object at position p is defined by the type of relationship which has the largest weight.

$$s_p = \sum_k w_{type(rel(i,j_k))} \tag{1}$$

$$P(p|R_i) \propto \exp(-s_p) \tag{2}$$

$$r_p = r_{p,k'} \quad k' = \arg\max\{w_{type(rel(i,j_k))}\} \tag{3}$$

A.3 Comparison between CSSG and advanced indoor scene generation algorithms

In Table 5, we summarize the properties of CSSG and other indoor scene algorithms. As the table shown, the state-of-the-art scene generation algorithms use SUNCG dataset [26] as training , is not currently not available. It is hard to reproduce the results from those approaches. In LUMINOUS, we reproduce the learning based approaches such as 3D-SLN [20] using publicly available dataset (3D-FRONT [40]) for training. We believe this could serve as first step to provide a unified benchmark for comparing indoor scene generation algorithms.

Algorithm	Scene graph Inference?	Constrained?	RGBD rendering?	Dataset?
PlanIT (2019)	✓	✓	✓	**unavailable**
Grains (2018)	**N/A**	**N/A**	✓	**unavailable**
3D-SLN (2020)	**N/A**	✓	✓	**unavailable**
Human-centric (2019)	**N/A**	**N/A**	✓	**unavailable**
Luminous CSSG	✓	✓	✓	**N/A**

Table 5: Comparison of CSSG and state-of-the-art indoor scene generation algorithms. Scene graph inference refers to the algorithm's ability to infer the latent scene graph of the indoor scene. Some of the algorithms support taking scene graphs as constraints. The dataset for training the indoor scene synthesis model is missing due to legal issues.

A.4 Implicit relationships between furniture

We list the implicit relationships when sampling the position of the furniture. Basically, the relationships can categorizes into two types: furniture v.s. room structure, and furniture v.s. furniture.

High-level action	Instruction candidates
GotoLocation	go to, find, walk to
PickupObject	pick up, take, carry
PutObject	put, place
SliceObject	slice, cut
CoolObject	chill, cool
HeatObject	heat, cook
CleanObject	clean, wash, rinse
ToggleObject	turn on

Table 6: Language template: mapping high-level actions to language instructions

- furniture v.s. room structure: (CounterTop, against, wall border), (TVStand, against, wall border), (Sofa, against, wall border), (border, against, wall border), (Bed, against, wall border), (Dresser, against, wall border),(Desk, against, wall border),(SideTable, against, wall border),(FloorLamp, against, wall corner), (DiningTable, away from, wall border)
- furniture v.s. furniture: (Chair, face, Desk), (Stool, face, DiningTable), (CoffeeTable, beside, Sofa), (DiningTable, away from, Sofa)

If multiple relationships influence the distribution of the sampling position of an object, we give the weight coefficient as 2.0 if the relationship is from *furniture v.s. room structure*, and as 1.0 if the relationship is from *furniture v.s. furniture*.

B Task Instructions Generation

Unlike ALFRED, LUMINOUS obtains the natural language as high-level instructions from an automatic pipeline instead of human annotations.

We design a language template to generate natural language instructions corresponding to the high-level instructions in ALFRED. Table 6 shows mappings from high-level action to language instructions. The natural language instruction is generated as:

$$[instruction\ candidate] + [object\ name] + [attribute]$$

Where the attribute specifies the receptacle for *PickupObject* (e.g., pick up an apple *in the fridge*), or the target location for *PutObject* (e.g., put a book *on the table*).

However, the language instruction for navigation can be too simple and vague if we just say *go to* some place. We apply the *Speaker* provided by ET to generate task instructions, especially for the navigation part. The training data come from the ALFRED dataset. The input of the *Speaker* is the low -level action sequence (e.g. *MoveAhead*, *MoveAhead*,*RotateLeft*) and images from the egocentric view the agent, and the output is a piece of natural language instruction.

$$(low\ level\ actions, images) \xrightarrow[\text{Speaker}]{} (language\ instructions)$$

We refer readers to ET [34] for model details and put the generated examples in Appendix C

C Illustration of ALFRED and LUMINOUS

In this part, we illustrate the details when we apply LUMINOUS for ALFRED challenge.

C.1 Task parser

The task parser is applied to deduce the indoor scene description I_i for an ALFRED trajectory T_i. Specifically, the task parser would go through the low-level actions in T_i, and

- extract the $action\ args$ as required objects from actions including *GotoLocaiton*, *PickupObject*, *ToggleObjectOn*, and *OpenObject*. For example, if the $action\ args$ of *GotoLocaiton* is *DiningTable*, the task parser put *DiningTable* into the list.

Figure 6: Living rooms sampled by LUMINOUS

Figure 7: Bedrooms sampled by LUMINOUS

- extract the $action\ args$ of *PickupObject* as scene constraints. For example, picking up an apple on the fridge means that initially *Apple* is in the *Fridge*.

C.2 Indoor scene sampling

For room structures of living rooms and bedrooms, LUMINOUS only keep *wall*, *ceiling*, *floor*, *window* and *door*. For room structures of kitchens and bathrooms, LUMINOUS further keeps *CounterTop*, *Sink*, *Cabinet*, and *Oven*, and *Bathtub*. Figure 6, 7, and 8 plot the scenes of different room types sampled by LUMINOUS.

C.3 ALFRED trajectories v.s. LUMINOUS trajectories

We performance side by side comparison between ALFRED trajectories and LUMINOUS trajectories in Figure 9 and 10. We plot the scene layouts, initial camera images, images after task completion and language instructions for both.

C.4 Hard task analysis: Heat & Place / Cool & Place

We notice the low success rate for two types of tasks: *Heat & Place* and *Cool & Place* in LUMINOUS scenes. The *Cool* operation requires a fridge and the *Heat* operation needs a microwave. We compare

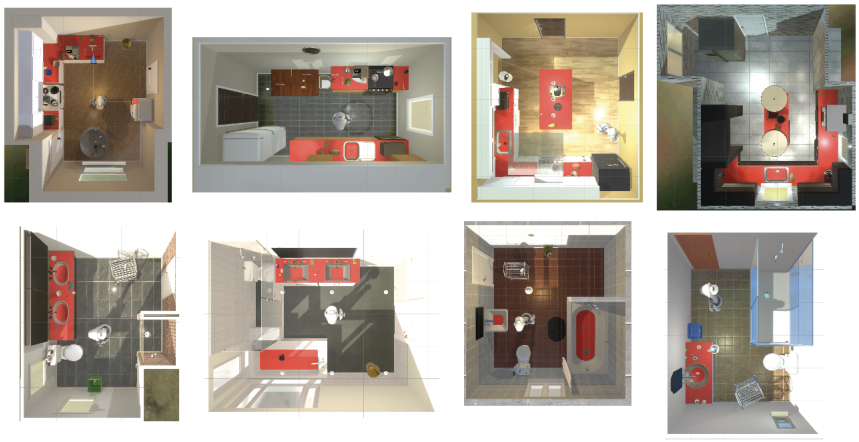


Figure 8: Kitchens and bathrooms sampled by LUMINOUS

Figure 9: Comparison between ALFRED and LUMINOUS generated trajectories. Task name: look_at_obj_in_light-AlarmClock-None-DeskLamp; scene name: FloorPlan301_physics; trial id:trial_T20190907_174127_043461.

the layout w.r.t. the fridge and microwave between AI2Thor scenes and LUMINOUS scenes, and we find a somewhat different set-up for them. Figure 11 compares the locations of the fridge and microwave. Since AI2Thor scenes are manually designed.

- In the task sampling stage (Table 2), the FF-Planner samples task trajectories from ground-truth knowledge of the environment and would not be influenced by visual discrepancies between ALFRED and LUMINOUS.
- In the EAI evaluation stage (Table 4), the EAI agent takes the input as RGB images and images look visually different between manually designed scenes and synthesized scenes, making the agent harder to complete heat and cool tasks.

D Large-Scale Evaluation Experiments

In this section, we conduct an additional large-scale evaluation with respect to the number of scenes. We generated 216 scenes with the same room structure as training scenes in ALFRED (including walls, floor, and windows) but randomized layouts and objects as the evaluation environments for ALFRED-like tasks. We summarize the statistics of our evaluation datasets and performance of

Figure 10: Comparison between ALFRED and LUMINOUS generated trajectories. Task name: pick_and_place_simple-Box-None-Sofa-205; scene name: FloorPlan205_physics; trial id:trial_T20190907_214755_478301.

Figure 11: Different locations of the microwave and fridge in AI2Thor scenes and LUMINOUS scenes. In AI2THOR, most microwaves and fridges are embedded in the structure of the room; in LUMINOUS, microwaves are preferred to be placed on a countertop and fridges most likely locates in a relatively open area. Such difference brings different visual experience to EAI agents.

Challenge	Navigation?	Interaction?	Language understanding?	affordance/physics understanding?
ObjectNav (habitat, ai2thor)	YES	NO	NO	NO
Multi-On/Rearrangement (habitat, ai2thor)	YES	PART OF	NO	YES
InteractiveNav (iGibson)	YES	YES	NO	YES
ALFRED (ai2thor)	YES	YES	YES	YES

Table 7: Comparison between ALFRED with other EAI tasks. Different simulators may have different requirements to EAI agents including navigation (to navigate an agent from one place to another), interaction (to interact with an object in the environment), language understanding (to follow language instructions from users), and affordance or physics understanding (to gain some knowledge for the affordance map in the scene).

three state-of-the-arts in Table 8. The second column presents the number of unique configurations (including room layouts, small object locations) of tasks in each task type. The third column shows the number of unique scenes/layouts (same room layout with different small object locations count as the same scene). Comparing the results in Table 4 and Table 8, the success rate in $S+$ column evaluated by 40 scenes and 216 scenes maintain the similar relative performance. Based on the above observation, we further strengthen our conclusions obtained in Section 4.2 that LUMINOUS can provide more robust and consistent evaluation results.

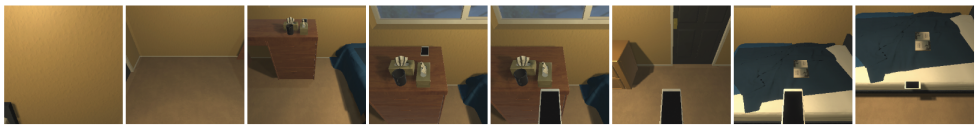
			ALFRED Inference Model					
			MOCA		**ET**		**HiTUT**	
Task	**# Trajs**	**# Scenes**	S	S+	S	S+	S	S+
Pick	1124	192	.295	.139	.500	.205	.359	.296
Cool	885	44	.261	.000	.532	.009	.190	.043
Stack	1002	126	.052	.002	.296	.028	.122	.058
Heat	786	54	.158	.000	.458	.005	.140	.061
Clean	923	98	.223	.000	.482	.109	.500	.232
Examine	1263	84	.202	.000	.426	.056	.266	.124
Pick Two	944	168	.112	.013	.419	.034	.177	.097
Overall	7074	-	.186	.025	.448	.068	.252	.137

Table 8: **Success rate on ALFRED tasks.** # Trajs: number of unique task configurations; # Scenes: number of unique scene layouts in each task type; S: ALFRED *seen*; S+ *Seen Plus via* LUMINOUS.

E Dataset Examples

The dataset examples from LUMINOUS are shown in Figure 12 and Figure 13.

Pick & Place

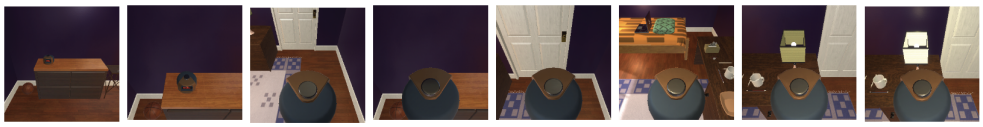


Alfred task reference: pick_and_place_simple-CellPhone-None-Bed-313 trial_T20190907_092221_700361

Goals: pick up the cellphone and place it on the bed

Instructions: "turn around and walk to the end of the room , then turn right and walk to the end of the room , then turn right and walk to the end of the room ."
"pick up the cellphone"
"turn around and walk to the end of the room , then turn left and walk to the end of the bed ."
"put the phone on the bed."

Examine in Light

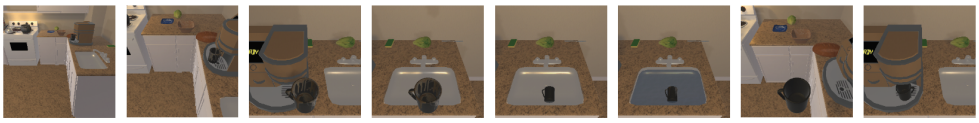


Alfred task reference: look_at_obj_in_light-AlarmClock-None-DeskLamp-301/trial_T20190907_174127_043461

Goals: look at alarm clock under the light

Instructions: "turn left and walk to the small black table in the corner ."
"take the alarm clock."
"turn around and walk to the white table on the left ."
"turn on the lamp on the table"

Clean & Place

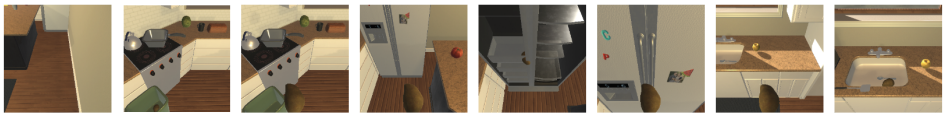


Alfred task reference: pick_clean_then_place_in_recep-Mug-None-CoffeeMachine-15/trial_T20190909_111443_363349

Goals: Clean a mug and put it in the coffee machine

Instructions: "walk forward , then turn right to face the coffee table ."
"carry the mug."
"turn right and walk to the sink ."
"wash the mug in the sink."
"turn left and walk to the coffee maker ."
"put the mug in the coffee maker ."

Cool & Place



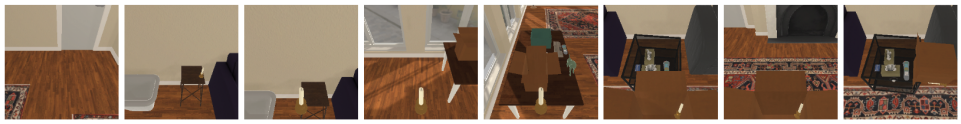
Alfred task reference: pick_cool_then_place_in_recep-Potato-None-SinkBasin-30/trial_T20190906_210247_069914

Goals: Cool the potato and place it on the sink

Instructions: "turn left and walk to the end of the kitchen counter , then turn right and walk to the end of the room , then turn right and walk to the end of the sink ."
"carry the potato."
"turn right and walk to the end of the room , then turn right and walk to the end of the room , then turn right and walk to the end of the room , then turn right and face the room ."
"cool the potato in the fridge then take it back out."
"turn around and walk to the sink ."
"put the potato in the sink"

Figure 12: Dataset Examples. Automatically generated scenes, low-level actions, and language instructions.

Stack & Place



Alfred task reference: pick_and_place_with_movable_recep-Candle-Box-CoffeeTable-230/trial_T20190906_232543_804101

Goals: Place the box with candle in it on the coffee table

Instructions: "turn around and walk to the other side of the room , then turn right and walk to the end of the room , then turn right and walk to the end of the room ."
"carry the candle."
"turn around and walk back to the other side of the room , then turn left and walk to the end of the room , then turn right and walk to the end of the room ."
"put the candle on the box."
"carry a box."
"turn around and walk to the coffee table in front of the couch ."
"Place the box with candle in it on the coffee table"

Heat & Place

Alfred task reference: pick_and_place_with_movable_recep-Candle-Box-CoffeeTable-230/trial_T20190906_232543_804101

Goals: Heat the apple and place it on the sink

Instructions: "turn right and walk to the end of the room , then turn right and walk to the sink ."
"carry the apple."
"turn left and walk to the microwave ."
"cook the apple in the microwave then take it back out."
"turn right and walk to the sink ."
"put the apple in the sink ."

Pick Two & Place

Alfred task reference: pick_two_obj_and_place-Newspaper-None-Sofa-218/trial_T20190907_203939_531678

Goals: Put two newspaper on the sofa

Instructions: "turn right and walk to the end of the coffee table , then turn left and walk to the end of the table , then turn left and walk to the table ."
"carry a newspaper."
"turn around and walk to the other side of the coffee table ."
"place the newspaper on the sofa."
"turn right and walk to the end of the room , then turn right and walk to the end of the room , then turn left and walk to the end of the room ."
"carry a newspaper."
"turn around and walk to the other side of the couch ."
"put the newspaper on the couch"

Figure 13: Dataset Examples. Automatically generated scenes, low-level actions, and language instructions.

F Related Work

LUMINOUS builds on and extends research in indoor scene synthesis, simulation environments in EAI, and language-guided task completion.

Indoor Scene Synthesis. In computer graphics, extensive research exists in 3D indoor scene synthesis. Early work either used explicit rule-based constraints [25] or incorporated stochastic priors into the generative procedure [15, 16, 17, 18]. Recent advances [19, 20, 22] utilize deep neural networks to extract patterns from large-scale datasets [26]. While these data-driven approaches significantly enhance the automation of the scene generation process, the resulting synthesized scenes are still relatively simple in terms of object quantity and inter-object spatial relationships. Many works generate scenes based on the natural representation of the scene graph [21, 19, 20]. Other lines of research condition on the image [24, 27] or text [28, 29] representation of indoor scenes. The discrepancies in the input representation of scene generation models and the diverse sources of data make it difficult to compare and contrast the performance of different methods. To facilitate research in learning-based approaches, LUMINOUS is designed to support end-to-end scene generation evaluation and a unified rendering tool to accommodate the outputs of various approaches simultaneously.

Embodied AI Simulators. In the past few years, researchers have developed many simulation environments [6, 7, 13, 5, 9] to serve as training and evaluation platforms for embodied agents. These simulation environments propel research progress in a wide range of embodied tasks, including vision-and-language task completion [10, 30], rearrangement [12, 7], navigation [9, 13], manipulation [31, 32] and human-robot collaboration [5]. Recently, AllenAct [33] integrates a set of embodied environments (such as iThor, RoboThor, Habitat [9], etc.), tasks, and algorithms thereby facilitating the evaluation of the same model or algorithm across multiple EAI platforms. Many EAI platforms are designed with sophisticated indoor scenes to perform embodied tasks. Platforms such as iGibson [13], AI2Thor [6] can randomize materials, color, and small objects in the scene, while the basic room layouts remain unchanged. To facilitate more robust and thorough evaluation of embodied agents, LUMINOUS automatically generates indoor scenes with randomized layouts at a large scale that readily support vision-and-language navigation and high-level object interactions. We summarized the properties of LUMINOUS and most popular EAI simulation platforms in Table 9.

Language-Guided Task Completion. Among existing EAI challenges, we use ALFRED [10] as our downstream exemplar task to evaluate the scene generation quality of LUMINOUS. ALFRED enables agents to follow natural language descriptions to complete complex household tasks. ALFRED tasks involve resolving vision-and-language grounding, affordance-aware navigation, and high-level object interactions. Roughly speaking, there are two categories of approaches to tackling ALFRED. Initial approaches learned end-to-end models that mapped language instructions into low-level actions directly [30, 3, 34]. Subsequently, hierarchical approaches [4, 35] were proposed that enabled better generalization and interpretation. However, those approaches are only tested in four indoor scenes unseen during training time. Towards a more convincing evaluation, LUMINOUS generates an order of magnitude larger number of scenes for better assessment of generalization and robustness.

Simulator	Layout randomization	Small Object randomization	Object material randomization	Number of scenes/rooms	Number of objects
Habitat (2020)	**N/A**	**N/A**	**N/A**	120	N/A
Virtualhome (2019)	**N/A**	**N/A**	✓	7(house)	357
threeDworld (2021)	**N/A**	✓	✓	100+	1000+
iGibson (2021)	**N/A**	✓	**N/A**	106(house)	1984
AI2Thor (2021)	**N/A**	✓	✓	227	2000+
Luminous	✓	✓	✓	∞	**2000+**

Table 9: Comparison of LUMINOUS and existing embodied AI simulation platforms. Layout randomization specifies the simulator's ability to change the furniture layout; small object randomization refers to change the layout of items on an affordance such as table and countertop; object material randomization changes the texture and color of an object.