

Towards a Feedback-driven Adaptive Embodied Environment Generator for Training Robotic Agents

Teresa Yeo^{1*}, Dulaj Weerakoon^{1*}, Dulanga Weerakoon¹, Archan Misra²

¹Singapore-MIT Alliance for Research and Technology Centre

²Singapore Management University

teresa.yeo@smart.mit.edu, dulaj.weerakoon@smart.mit.edu, dulanga.weerakoon@smart.mit.edu, archanm@smu.edu.sg

Abstract

Training embodied robotics agents often require large corpora of training data, which in current approaches are manually created using synthetic environment generators. Although such synthetic generators allow training agents across a variety of embodied tasks, generating sufficiently large datasets remains a challenge due to the extensive manual effort involved. To address this limitation, we present an early prototype of an automated approach for generating synthetic environments for training robotic agents. The proposed environment generator is driven by feedback from the agent as it performs the embodied task within the environment, allowing the generation process to adapt dynamically based on the agent’s competence. This feedback-driven automation enables the creation of significantly larger and more diverse corpora of training data for embodied agents across a wide range of environmental conditions.

Introduction

Embodied Artificial Intelligence (Embodied AI) represents a critical step toward achieving Artificial General Intelligence (AGI). In this domain, disembodied agents refer to systems that operate entirely within virtual environments. Examples include Apple Siri (Apple 2025), Amazon Alexa (Amazon 2025), and recent Large Language Model (LLM) based systems such as ChatGPT (OpenAI 2022), which can perceive and reason about the physical world only through symbolic or linguistic representations while remaining constrained to virtual settings. In contrast, embodied robotic agents are capable not only of perceiving and reasoning about physical environments but also of navigating and performing actions within complex and dynamic real-world contexts.

Disembodied agents, typically equipped with Vision Language Models (VLMs), are trained on internet scale datasets and therefore exhibit strong reasoning and conversational capabilities. However, the same does not hold for embodied agents, as comparable large scale datasets are not available for their training. Very recently, the reasoning and planning abilities of embodied agents are being supported by Vision Language Action models (VLAs), which map an embodied task prompt, typically expressed as a natural language

instruction together with the continuous RGB stream from the robot’s camera, into a sequence of low-level robotic actions. Training such VLA models requires tuples of [embodied language prompt, RGB, action sequence], which are not present in internet data and therefore remain far more limited in scale compared to the datasets used for training VLMs.

Training embodied agents presents an additional layer of complexity due to the wide variation in their design and functionality across different use cases. For example, some agents are designed purely for navigation, whereas others incorporate manipulators for tasks such as picking and placing objects. As a result, the action space varies considerably across embodied agents, requiring the underlying VLA model to be customized for each specific configuration and task setting.

Amid these challenges, data scarcity remains a fundamental bottleneck in training embodied robotic agents. To address this, several works such as AI2-THOR (Kolve et al. 2017) and Genesis (Zhou et al. 2024) have explored the use of synthetic embodied environments to train agents under diverse and controlled conditions. While such synthetic environments enable large-scale data generation, they still demand substantial manual effort to design and curate sufficiently diverse scenarios. Consequently, ensuring that agents can generalize across a broad range of real-world conditions remains difficult, ultimately constraining the scalability of these synthetic data generation approaches.

To tackle these challenges, we propose an early concept towards developing an automated embodied environment generator for training robotic agents. A key feature of our generator is that the environment generation process is conditioned on feedback from the agent operating within the environment, allowing it to iteratively and automatically adapt to the agent’s competence level. This feedback-based mechanism enables *procedural and adaptive curriculum generation* for training embodied robotic agents. Through this automated approach of embodied environment generation, we address the challenge of data scarcity in training accurate and robust agents for a variety of robotic tasks.

Method

We now describe our proposed method illustrated in Fig. 1.

*These authors contributed equally.

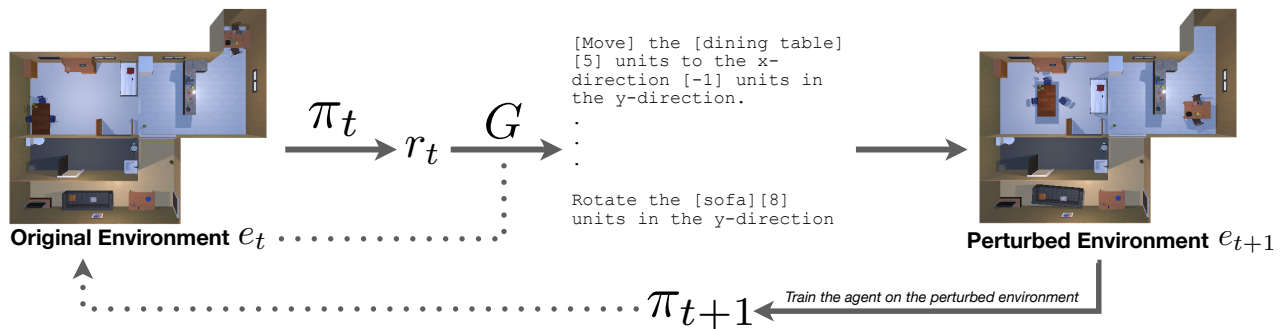


Figure 1: **Overview of the proposed adaptive environment generation framework.** An agent performing embodied tasks such as object navigation is first evaluated to obtain a reward r_t . This reward, together with the current environment configuration e_t , is provided to a generator G , which proposes a sequence of object-level modifications to increase task difficulty. The resulting, more challenging environment is then used to retrain the agent, completing the feedback loop between environment generation and agent learning.

Problem formulation. Let $e \in \mathcal{E}$ denote an environment configuration, represented by a structured scene graph or parameterized layout. An agent with policy $\pi(a | s)$ performs embodied tasks such as object navigation, exploration, or manipulation to obtain a reward signal $r = R(\pi, e)$.

We define an environment generator $G : \mathcal{E} \times \mathcal{R} \rightarrow \mathcal{E}$, which takes as input the current environment configuration e_t and the agent’s recent reward r_t , and outputs a new configuration $e_{t+1} = G(e_t, r_t)$. The goal of G is to produce environments that are both *sufficiently* challenging and diverse for the agent. Formally, G aims to *maximize* the following objective $\mathcal{J}(G) = E_t [\Delta R(\pi_t, G(e_t, r_t))]$ where ΔR measures the improvement in agent performance on held-out environments. This formulation establishes a closed feedback loop between the generator and the agent, allowing for *procedural and adaptive curriculum generation*.

Structured representation of the environment. The environment configuration e can be represented as a structured graph or parameterized layout capturing the composition of the scene. One such representation is $e = (O, A, R)$, where $O = \{o_i\}$ denotes the set of objects, $A(o_i)$ the attributes of each object (e.g., position, rotation, scale, material), and $R(o_i, o_j)$ the pairwise spatial or functional relations (e.g., on, next-to, inside). This representation can be instantiated in simulators such as AI2-THOR, where an environment is defined by a configuration file specifying object types and their parameters, effectively forming a directed scene graph.

Such a structured representation enables G to perform *compositional modifications*—adding, removing, or perturbing objects or relations. Furthermore, it also enables a separate validation module to check physical consistency and task feasibility (e.g., ensuring that objects do not overlap and tasks remain solvable).

Adaptive environment generation. Given the agent’s policy π_t and its reward signal $r_t = R(\pi_t, e_t)$ from the environment e_t , the generator G aims to produce a more challenging environment configuration $e_{t+1} = G(e_t, r_t)$ for the agent. Concretely, G aims to generate an $e_{t+1} = \arg \min_{e' \in \mathcal{E}} R(\pi_t, e')$ subject to validity and task solvability constraints. The agent is then trained in the new environment e_{t+1} . This results in an updated policy π_{t+1} . This interaction

between the agent and the generator results in a *curriculum* where the generator continually synthesizes harder environments that are *adapted* to the agent’s competence.

The generator G can be instantiated in multiple ways depending on the desired level of structure and interpretability. One approach leverages an LLM that conditions on the agent’s trajectory or top-down map of its behavior in e_t , and outputs a sequence of discrete editing actions—e.g., selecting an object $o_i \in O$ and modifying its spatial parameters or relations to create a new configuration e_{t+1} . As LLMs possess spatial reasoning and world-model priors, this formulation enables broad generalization to unseen scenes and tasks, often producing semantically coherent and meaningful modifications without task-specific training. In contrast, G_θ can be trained to predict configuration deltas given e_t and the reward r_t . This approach supports gradient-based optimization, lower latency, and scalability across many agents or tasks, but may require task-specific data and tends to generalize less reliably beyond the training distribution.

Conclusion

This work proposes a feedback-driven framework, where a generator adaptively modifies simulation environments in a *neuro-symbolic manner* (combining structured perturbations and reinforcement learning) based on an agent’s performance. By coupling environment generation and agent retraining, the system produces challenging scenarios that help embodied agents generalize and become robust.

Acknowledgments

This research is supported by the National Research Foundation (NRF), Prime Minister’s Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. The Mens, Manus, and Machina (M3S) is an interdisciplinary research group (IRG) of the Singapore MIT Alliance for Research and Technology (SMART) centre.

References

Amazon. 2025. Learn More about Alexa Features, Skills, and Products. <https://www.alex.com/>. Accessed November 7, 2025.

Apple. 2025. Siri. <https://www.apple.com/sg/siri/>. Accessed November 7, 2025.

Kolve, E.; Mottaghi, R.; Han, W.; VanderBilt, E.; Weihs, L.; Herrasti, A.; Deitke, M.; Ehsani, K.; Gordon, D.; Zhu, Y.; et al. 2017. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*.

OpenAI. 2022. Introducing ChatGPT. <https://openai.com/index/chatgpt/>. Accessed November 7, 2025.

Zhou, X.; et al. 2024. Genesis: A Generative and Universal Physics Engine for Robotics and Beyond. <https://github.com/Genesis-Embodied-AI/Genesis>.