LEGENT: Open Platform for Embodied Agents

Zhili Cheng [1] **Jinyi Hu** [1] **Zhitong Wang** [1] **Yuge Tu** [1] **Shengding Hu** [1] **An Liu** [1] **Pengkai Li** [1] **Lei Shi** [1] **Zhiyuan Liu** [1] **Maosong Sun** [1]

Abstract

Despite advancements in Large Multimodal Models (LMMs), their integration into language-grounded, human-like embodied agents remains incomplete, hindering complex real-life task performance in physical environments. Existing integrations often feature limited open sourcing, challenging collective progress in this field. We introduce LEGENT, an open, scalable platform for developing embodied agents using LMMs. LEGENT offers a dual approach: a rich, interactive 3D environment with communicable and actionable agents, paired with a user-friendly interface, and a sophisticated data generation pipeline utilizing advanced algorithms to exploit supervision from simulated worlds at scale. In our experiments, an embryonic vision-language-action model trained on LEGENT-generated data surpasses GPT-4V in embodied tasks, showcasing promising generalization capabilities.

1. Introduction

Large Language Models (LLMs) and Large Multimodal Models (LMMs) (Team et al., 2023; OpenAI, 2023), i.e., Multimodal Foundation Model, present inspiring capabilities in understanding and generating human-like text and realistic images. However, their direct application in embodied AI, where agents interact in physical or simulated environments, is still primitive. LMMs lack the necessary grounding (Harnad, 1990) in physical interactions to operate in these settings effectively.

Research in embodied intelligence has evolved significantly, leading to more realistic and sophisticated environments (Kolve et al., 2017; Puig et al., 2018; Savva et al., 2019; Puig et al., 2023b) and increasingly challenging tasks (Das et al., 2018; Gordon et al., 2018; Batra et al., 2020). However, these traditional environments and approaches are typically incompatible with current LMMs, which hinders the seamless integration of task execution via language interaction. Consequently, these approaches do not leverage the extensive generalizable knowledge present in LMMs.

To achieve generalizable embodied intelligence, two key factors are crucial: language grounding to utilize the extensive knowledge in LMMs and the training data for embodied AI at scale. There have been noteworthy efforts in combining embodied AI with LMMs (Reed et al., 2022; Brohan et al., 2023). They collect large-scale training data from embodied scenes and train end-to-end models that interpret both language and visual inputs and perform corresponding actions. However, the lack of open-source access to these environments and datasets restricts open-source community-wide progress in this field.

Towards this aspiration, we introduce LEGENT, an open-source and user-friendly platform that enables scalable training of embodied agents based on LMMs. LEGENT contains two parts. First, it provides a 3D embodied environment with the following features: (1) Diverse, realistic, and interactive scenes; (2) Human-like agents with egocentric vision capable of executing actions and engaging in direct language interaction with users; (3) User-friendly interface offering comprehensive support for researchers unfamiliar with 3D environments. Second, LEGENT builds a systematic data generation pipeline for both scene generation and agent behavior, incorporating state-of-the-art algorithms for scene creation (Deitke et al., 2022; Yang et al., 2023b) and optimal trajectory generation. In this way, extensive and diverse trajectories of agent behavior with egocentric visual observations and corresponding actions can be generated at scale for embodied agent training.

To demonstrate the potential of LEGENT, we train a basic vision-language-action model based on LMMs with generated data on two tasks: navigation and embodied question answering. The model processes textual and egocentric visual input and produces controls and textual responses directly. The prototype model outperforms GPT-4V (OpenAI, 2023), which lacks training in an embodied setting. The generalization experiment reveals the LEGENT-trained model's ability to generalize to unseen settings. LEGENT

[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China. Correspondence to: Maosong Sun <sms@tsinghua.edu.cn>.

Accepted at the 1st Machine Learning for Life and Material Sciences Workshop at ICML 2024.

platform and its documentation are publicly available at https://docs.legent.ai.

2. Related Work

Embodied Environment. Embodied environments are extensively utilized in games (Oh et al., 2016) and robotics (Kolve et al., 2017; Yan et al., 2018; Gan et al., 2020; Puig et al., 2023a), with a primary focus on visual AI and reinforcement learning. Some platform focuses on specific embodied tasks, such as manipulation (Yu et al., 2020; Makoviychuk et al., 2021), navigation (Chang et al., 2017; Dosovitskiy et al., 2017), or planning-oriented agents (Puig et al., 2018; Shridhar et al., 2020). However, existing platforms' setups fall short in accommodating the training of LMMs, which require diverse and large-scale supervised data to integrate embodied capability.

LMMs-based Embodied Agent. Noteworthy studies have concentrated on developing embodied models capable of end-to-end operation, as demonstrated in the works of Reed et al. (2022); Brohan et al. (2023); Belkhale et al. (2024). Despite being generalizable, the datasets and models in these studies are not publicly available.

Scene Generation. Scene generation has demonstrated significant effectiveness in training embodied agents by ProcTHOR (Deitke et al., 2022). Compared to employing manually crafted rules used in ProcTHOR, recent studies (Wen et al., 2023; Yang et al., 2023b; Feng et al., 2024) leverage prior knowledge of large language models and propose algorithms to generate diverse, high-quality scenes.

Agent Trajectory Generation. Some research focuses on crafting reward functions to guide small policy models (Yu et al., 2023; Xian et al., 2023; Wang et al., 2023; Ma et al., 2023). However, there will be huge costs and instability when applying reward-based training to large foundation models. Meanwhile, pioneering efforts have been made in code generation for robotics (Liang et al., 2023; Singh et al., 2023; Vemprala et al., 2023; Huang et al., 2023) and trajectory generation for imitation learning (Kamath et al., 2023). We follow this paradigm and further scale agent trajectory generation to cover a wide range of tasks.

3. LEGENT

In this section, we introduce our platform LEGENT. The design of LEGENT involves scene, agent, and interface. All three components are specially tailored for the integration of LMMs and ensure scalability.

3.1. Scene

The design of the scenes in LEGENT emphasizes **interactivity** and **diversity**, striving for a versatile and scalable environment that enriches the training of embodied agents for wide application.

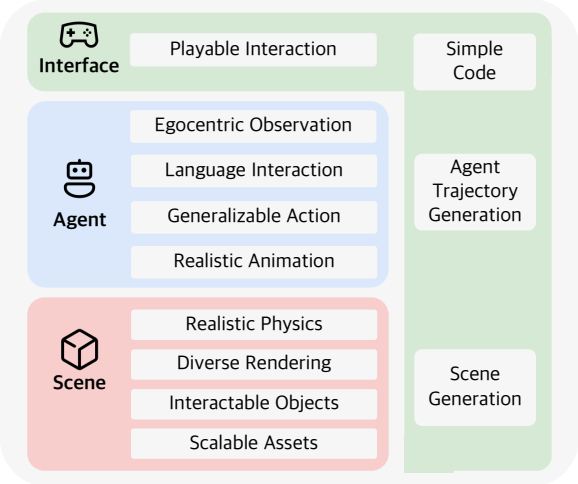

Figure 1. Features of LEGENT.

Realistic Physics. LEGENT provides a real-time simulation that closely mirrors real-world physics based on game engines. It supports realistic effects like gravity, friction, and collision dynamics, improving agents' embodied comprehension or aiding the development of generative world simulators (Yang et al., 2023a).

Diverse Rendering. Unlike the fixed stylized renderings in games and the photorealism in robotics, LEGENT integrates these styles by customizing rendering functions, which allows easy transitions between rendering styles to accommodate different requirements for flexible usage.

Interactable Objects. In LEGENT, both agents and users can manipulate various fully interactable 3D objects, which enables actions such as picking up, transporting, positioning, and handing over these objects. Additionally, the environment supports interaction with dynamic structures, such as doors and drawers. We anticipate that the scope of these dynamic structures will be significantly broadened through the application of generative methods (Chen et al., 2023).

Scalable Assets. LEGENT supports importing customized objects, including user-supplied 3D objects, objects from existing datasets (Deitke et al., 2023) and those created by generative models (Siddiqui et al., 2023; Wang et al., 2024). We choose glTF as the import format for its openness and compatibility. This feature grants users the flexibility to customize the scene by strategically placing these assets or integrating them into scene generation algorithms.

3.2. Agent

The agent is designed with two criteria: emulating human interactions and compatibility with LMMs.

Egocentric Observations. Following the previous study for

interactive embodied agents (Team et al., 2021), the agent is equipped with egocentric vision. The egocentric vision is captured by mounting a camera on the agent's head.

Language Interaction. Users and agents can communicate with each other in natural language in LEGENT. Grounding language within the environment will potentially connect the extensive knowledge in LMMs with embodied experience.

Generalizable Actions. Agents in LEGENT are capable of performing a range of actions, including navigation, object manipulation, and communication. Regarding the instantiation of actions, existing literature can be broadly categorized into two types: *executable plans* (Puig et al., 2018; Shridhar et al., 2020) and *control* (Kolve et al., 2017; Savva et al., 2019). In *executable plans*, actions are expressed through sub-steps to complete a task, such as "*walk towards apple 1*", which requires an additional module for execution. *Control*, on the other hand, refers to the action expression like "*move forward 1 meter*". In LEGENT, we use *control*, targeting generalizing to new environments with real-world settings. The learned actions can be integrated with diverse actuators with the least additional effort.

Another important action design is allowing the agent to execute continuous actions such as moving forward across a continuous distance, as opposed to moving in a grid-by-grid manner. This design offers two advantages for LMM-based agents: (1) It minimizes the inference cost of LMMs by eliminating the need for constant frame-by-frame inference. (2) It addresses the issue of minimal information gain observed when agents move incrementally in a stepwise manner, a process that creates less effective data for model training.

Realistic Animation. LEGENT features precise humanoid animations using inverse kinematics and spatial algorithms, enabling lifelike movements. Also, when combined with egocentric vision, it offers a cost-effective alternative for immersive experiences similar to the Ego4D (Grauman et al., 2022), which requires a huge cost to collect at scale.

3.3. Interface

Our platform offers a user-friendly interface for researchers to integrate LMMs with the embodied environment easily, with little need for expertise in 3D environments. LEGENT is engineered to be cross-platform, ensuring smooth running on personal computers without demanding particular prerequisites or complex setups, and it facilitates connections to remote servers for training and deployment. The user interface of LEGENT is designed to be as intuitive as playing a video game with the agent within the environment, utilizing just a keyboard and mouse for navigation and interaction. This interface facilitates straightforward visual debugging and qualitative analysis. Besides, we provide a concise Python toolkit to enable the interaction between the agent and the environment and support scene generation and trajectory generation, which will be introduced in the following sections. Detailed guidance is available in our documentation.

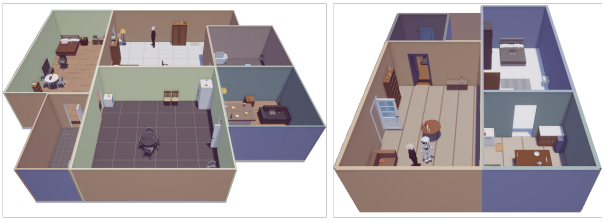

Figure 2. Generated scenes by ProcTHOR (L) and Holodeck (R).

4. Data Generation

The second part of LEGENT is a scalable data generation pipeline. It aims to exploit the inherent supervision from simulated worlds and support large-scale training of general-purpose embodied agents.

4.1. Scene Generation

Scene generation offers agents with diverse embodied experiences. LEGENT has currently integrated two scene generation methods: (1) Procedure generation efficiently creates large-scale scenes. (2) Language-guided generation captures the semantics of textual queries and leverages common sense knowledge to optimize spatial layouts.

Procedural Generation. We utilize the procedural scene generation proposed in ProcTHOR (Deitke et al., 2022), designed to create realistic indoor scenes at scale by integrating prior knowledge of object placement and spatial relationships. The implementation process starts with drafting a house layout, followed by the placement of large furniture, and ends with the arrangement of small objects. We provide an interface that allows users to input specific conditions for object occurrence and placement, enabling the generation of scenes tailored to specific tasks.

Language Guided Generation. We integrate Holodeck (Yang et al., 2023b) into LEGENT to generate indoor scenes given any natural language query. This process resembles procedural generation but is driven by LLMs instead of human-written programs. We ask LLMs to determine the exact locations of doors and floor objects, granting LLMs more control over the room layout.

4.2. Task Generation

We create diverse tasks expressed in language paired with specific scenes, thereby contextualizing each task within the environment. We employ the following two strategies.

Task Generation for Given Scenes. In this strategy, we

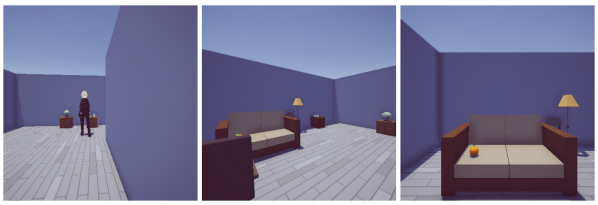

Figure 3. A generated trajectory for task "Where is the orange". The actions for the three observations are: 1. *rotate_right(-59)*; 2. *move_forward(1.2), rotate_right(-35)*; 3. *speak("It's on the sofa.")*.

serialize the generated scenes into a detailed textual description and present it to LLMs with crafted instructions. LLMs assume the role of human users, generating a variety of tasks. This approach is especially effective for generating diverse tasks automatically.

Scene Generation for Given Tasks. This approach efficiently generates large-scale samples for specific tasks, provided that the scene generation algorithm fulfills the required criteria. For instance, when the task involves querying an object's location, the algorithm generates a scene that includes the object and its receptacle, inherently creating question-answering annotations.

4.3. Trajectory Generation

Trajectories for training embodied agents comprise continuous sequences of egocentric observations and actions. The main challenge lies in accurately determining ground-truth actions for each step. Inspired by pioneering works in code generation for robotics, we utilize LLMs to write intermediate codes from provided state descriptions and instructions. These codes are instantiated as *multi-step controllers*, designed to calculate the optimal actions at each step given the internal states of the environment.

We demonstrate this process using an example task "*Where is the orange?*". As shown in Figure 3, to finish the task, the agent needs to search the room and answer the question. We ask LLMs to determine the object identifier of the orange in the scene and recognize its placement from the state description, thereby generating the following intermediate function "*find*": Then the code "*find*" is instantiated as a multi-step controller that utilizes pathfinding algorithms (Hart et al., 1968) incorporating visibility checks, which calculates the waypoints of the shortest path from the agent to the target object using a navigation mesh. The controller then calculates the controls of the agent to navigate along these waypoints. For instance, in the first observation shown in Figure 3, the agent needs to rotate 59 degrees to the left to orient to the next waypoint, resulting in the action "*rotate_right(-59)*". LEGENT records the visual observations and actions during this process as a trajectory, which can be exported as a video or an image-text interleaved sequence.

Task	Come Here		Where Is	
Room Num	One	Two	One	Two*
GPT-4V (zero-shot)	0.21	0.17	0.25	0.22
ViLA-7B-Sep 1K	0.87	0.28	0.30	0.22
ViLA-7B-Sep 10K	**0.96**	**0.70**	**0.94**	0.52
ViLA-7B-Joint	**0.96**	**0.70**	0.92	**0.65**

Table 1. Success rates on two tasks. *VILA-Sep* denotes models fine-tuned separately for each task, whereas *VILA-Joint* means models trained jointly on both tasks. * means generalization test.

4.4. Prototype Experiments

We conduct a prototype experiment to assess the utility of generated data on two embodied tasks: "Come Here" for social navigation (Puig et al., 2023a) and "Where Is" for embodied question answering (Das et al., 2018). Task complexity varied from navigating in one room to the more intricate two rooms. We generate 1k and 10k trajectories for the initial three tasks ("Come Here" in one or two rooms and "Where Is" in one room) and assess the models on 100 trajectories across all four tasks. The "Where Is" task in the two-room setting serves as a generalization test, which is not included in the training data.

Due to the lack of powerful video understanding models, we temporarily only focus on the observation at the end of each continuous action, formulating one trajectory as an image-text interleaved sequence. We utilize VILA-7B (Lin et al., 2023) as our backbone due to its capability in interleaved inputs. We train the model to output current action based on task descriptions and interleaved context of previous observations and actions.

The results presented in Table 1 lead to several observations: (i) GPT-4V struggles in these tasks, reflecting a lack of embodied experience in mainstream LMMs. (ii) Increasing training data improves the model performance. (iii) The navigational skills developed from the "Come Here" task in a two-room environment generalize well to the untrained task scenario, enhancing the model's ability to navigate in two rooms for the embodied question answering task. We leave the exploration of more large-scale training in the future work.

5. Conclusion and Future Work

In this work, we present LEGENT, an open platform for developing embodied agents, focusing on integrating LMMs with scalable embodied training. In our future releases, we prioritize: (1) Building a more diverse data generation pipeline. (2) Scaling model training. (3) Unifying humanoid animation with robotic control and refining the physics. (4) Improving scene generation and integrating 3D generation methods to support more diverse and realistic scenes.

References

Batra, D., Chang, A. X., Chernova, S., Davison, A. J., Deng, J., Koltun, V., Levine, S., Malik, J., Mordatch, I., Mottaghi, R., et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020.

Belkhale, S., Ding, T., Xiao, T., Sermanet, P., Vuong, Q., Tompson, J., Chebotar, Y., Dwibedi, D., and Sadigh, D. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

Chen, Q., Memmel, M., Fang, A., Walsman, A., Fox, D., and Gupta, A. Urdformer: Constructing interactive realistic scenes from real images via simulation and generative modeling. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., and Batra, D. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–10, 2018.

Deitke, M., VanderBilt, E., Herrasti, A., Weihs, L., Ehsani, K., Salvador, J., Han, W., Kolve, E., Kembhavi, A., and Mottaghi, R. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.

Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., and Farhadi, A. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153, 2023.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Feng, W., Zhu, W., Fu, T.-j., Jampani, V., Akula, A., He, X., Basu, S., Wang, X. E., and Wang, W. Y. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Gan, C., Schwartz, J., Alter, S., Mrowca, D., Schrimpf, M., Traer, J., De Freitas, J., Kubilius, J., Bhandwaldar, A., Haber, N., et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020.

Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., and Farhadi, A. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4089–4098, 2018.

Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

Harnad, S. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990.

Hart, P. E., Nilsson, N. J., and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

Huang, W., Wang, C., Zhang, R., Li, Y., Wu, J., and Fei-Fei, L. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Kamath, A., Anderson, P., Wang, S., Koh, J. Y., Ku, A., Waters, A., Yang, Y., Baldridge, J., and Parekh, Z. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10813–10823, 2023.

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

Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., and Zeng, A. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500. IEEE, 2023.

Lin, J., Yin, H., Ping, W., Lu, Y., Molchanov, P., Tao, A., Mao, H., Kautz, J., Shoeybi, M., and Han, S. Vila: On pre-training for visual language models. *arXiv preprint arXiv:2312.07533*, 2023.

Ma, Y. J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., and Anandkumar, A. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

Oh, J., Chockalingam, V., Lee, H., et al. Control of memory, active perception, and action in minecraft. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2016.

OpenAI, t. Gpt-4v(ision) system card, 2023.

Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., and Torralba, A. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8494–8502, 2018.

Puig, X., Undersander, E., Szot, A., Cote, M. D., Partsey, R., Yang, J., Desai, R., Clegg, A. W., Hlavac, M., Min, T., Gervet, T., Vondrus, V., Berges, V.-P., Turner, J., Maksymets, O., Kira, Z., Kalakrishnan, M., Malik, J., Chaplot, D. S., Jain, U., Batra, D., Rai, A., and Mottaghi, R. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023a.

Puig, X., Undersander, E., Szot, A., Cote, M. D., Yang, T.-Y., Partsey, R., Desai, R., Clegg, A. W., Hlavac, M., Min, S. Y., et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023b.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9339–9347, 2019.

Shridhar, M., Yuan, X., Côté, M.-A., Bisk, Y., Trischler, A., and Hausknecht, M. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Siddiqui, Y., Alliegro, A., Artemov, A., Tommasi, T., Sirigatti, D., Rosov, V., Dai, A., and Nießner, M. Meshgpt: Generating triangle meshes with decoder-only transformers. *arXiv preprint arXiv:2311.15475*, 2023.

Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., and Garg, A. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530. IEEE, 2023.

Team, D. I. A., Abramson, J., Ahuja, A., Brussee, A., Carnevale, F., Cassin, M., Fischer, F., Georgiev, P., Goldin, A., Gupta, M., et al. Creating multimodal interactive agents with imitation and self-supervised learning. *arXiv preprint arXiv:2112.03763*, 2021.

Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Vemprala, S., Bonatti, R., Bucker, A., and Kapoor, A. Chatgpt for robotics: Design principles and model abilities. *arXiv preprint arXiv:2306.17582*, 2023.

Wang, Y., Xian, Z., Chen, F., Wang, T.-H., Wang, Y., Fragkiadaki, K., Erickson, Z., Held, D., and Gan, C. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *arXiv preprint arXiv:2311.01455*, 2023.

Wang, Z., Wang, Y., Chen, Y., Xiang, C., Chen, S., Yu, D., Li, C., Su, H., and Zhu, J. Crm: Single image to 3d textured mesh with convolutional reconstruction model. *arXiv preprint arXiv:2403.05034*, 2024.

Wen, Z., Liu, Z., Sridhar, S., and Fu, R. Anyhome: Open-vocabulary generation of structured and textured 3d homes. *arXiv preprint arXiv:2312.06644*, 2023.

Xian, Z., Gervet, T., Xu, Z., Qiao, Y.-L., Wang, T.-H., and Wang, Y. Towards generalist robots: A promising paradigm via generative simulation. *arXiv preprint arXiv:2305.10455*, 2023.

Yan, C., Misra, D., Bennnett, A., Walsman, A., Bisk, Y., and Artzi, Y. Chalet: Cornell house agent learning environment. *arXiv preprint arXiv:1801.07357*, 2018.

Yang, M., Du, Y., Ghasemipour, K., Tompson, J., Schuurmans, D., and Abbeel, P. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023a.

Yang, Y., Sun, F.-Y., Weihs, L., VanderBilt, E., Herrasti, A., Han, W., Wu, J., Haber, N., Krishna, R., Liu, L., et al. Holodeck: Language guided generation of 3d embodied ai environments. *arXiv preprint arXiv:2312.09067*, 2023b.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Yu, W., Gileadi, N., Fu, C., Kirmani, S., Lee, K.-H., Arenas, M. G., Chiang, H.-T. L., Erez, T., Hasenclever, L., Humplik, J., et al. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*, 2023.