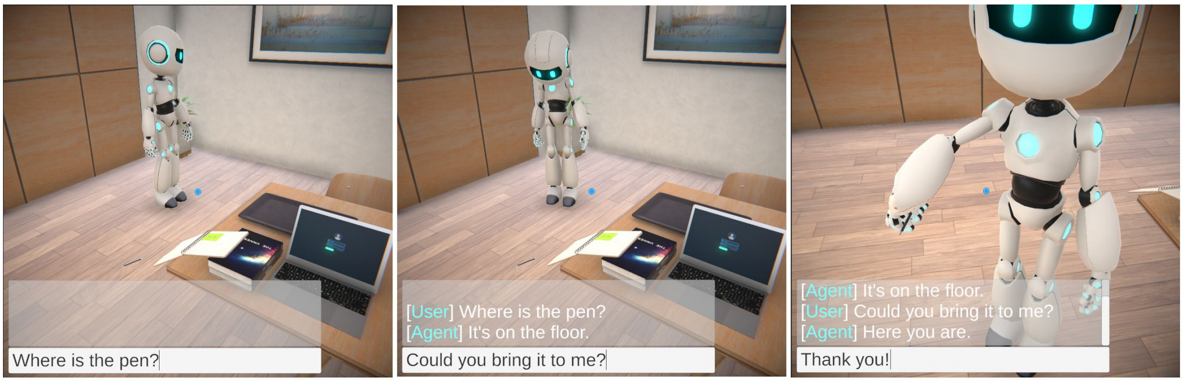
LEGENT: Open Platform for Embodied Agents

Zhili Cheng, Zhitong Wang, Jinyi Hu, Shengding Hu, An Liu,
Yuge Tu, Pengkai Li, Lei Shi, Zhiyuan Liu, Maosong Sun[*]
Department of Computer Science and Technology, Tsinghua University
{chengzl22, wangzt23, hu-jy21, hsd23}@mails.tsinghua.edu.cn
https://legent.ai



Where is the pen?

[User] Where is the pen?
[Agent] It's on the floor.

Could you bring it to me?

[Agent] It's on the floor.
[User] Could you bring it to me?
[Agent] Here you are.

Thank you!

Figure 1: Interaction with the embodied agent in LEGENT. These sequential interactions showcase the agent's ability to answer the user's questions and follow the user's instructions.

Abstract

Despite advancements in Large Language Models (LLMs) and 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 LLMs and 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. LEGENT is available at https://legent.ai.

1 Introduction

Large Language Models (LLMs) (Brown et al., 2020; Achiam et al., 2023; Touvron et al., 2023a,b) and Large Multimodal Models (LMMs) (OpenAI, 2023; Team et al., 2023; Liu et al., 2024; Hu et al., 2024) 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. LLMs and 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; Yenamandra et al., 2023). However, these traditional environments and approaches are typically incompatible with current LLMs and LMMs, which hinders the seamless integration of task execution via language interaction. Consequently, these approaches do not leverage the extensive generalizable knowledge present in LLMs and LMMs.

To achieve generalizable embodied intelligence, two key factors are crucial: language grounding to utilize the extensive knowledge in LMMs, and the expansion of training data for embodied AI. 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 mod-

[*] Corresponding author. Email: sms@tsinghua.edu.cn

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 335–345
August 11-16, 2024 ©2024 Association for Computational Linguistics

els 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. Therefore, the academic community urgently requires an open-source platform that facilitates the integration of language grounding in embodied environments and schemes to generate large-scale training data for embodied agents based on LLMs and LMMs.

Towards this aspiration, we introduce LEGENT, an open and user-friendly platform that enables scalable training of embodied agents based on LLMs and 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 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 platform and its documentation are publicly available at `https://legent.ai`.

2 Related Work

Embodied Environment. Embodied environments are extensively utilized in games (Johnson et al., 2016; Oh et al., 2016; Beattie et al., 2016) and robotics (Kolve et al., 2017; Yan et al., 2018; Xia et al., 2018; Gan et al., 2020; Li et al., 2021; 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; Wang et al., 2022). However, the environment setups and data frameworks of existing platforms fall short in accommodating the training of LMMs. LMMs excel in the supervised learning paradigm and necessitate diverse and large-scale data to integrate embodied capability. Existing platforms are not yet ready to scale, including: the primarily supported reinforcement learning methods require careful reward engineering, the diversity of the training data cannot be easily expanded, and collecting data for imitation learning on these platforms requires manual effort. Refer to Table 1 for a comparison of LEGENT with other embodied AI platforms.

LMMs-based Embodied Agent. Based on the development of LLMs and LMMs, researchers are endeavored to build agents for automatically completing human's instruction (Yao et al., 2023; Liu et al., 2023). For embodied tasks, existing studies have concentrated on developing embodied agents capable of end-to-end operation, as demonstrated in Reed et al. (2022); Brohan et al. (2023); Belkhale et al. (2024). However, 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 LLMs 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 (Garrett et al., 2021; Kamath et al., 2023; Dalal et al., 2023). These efforts align with our approach to generating large-scale embodied trajectories for training LMMs.

Platform	Functionality				Usability		Scalability			
	Real.	Anim.	Inter.	Lang.	Access	Cross.	Scene	Asset	Style	**Data**
Minecraft			✓	✓	✓	✓	✓			
AI2THOR (Kolve et al., 2017)	✓				✓					
Habitat (Savva et al., 2019)	✓				✓					
Playhouse (Abramson et al., 2020)			✓	✓			✓			
ProcTHOR (Deitke et al., 2022)	✓				✓		✓			
Habitat 3.0 (Puig et al., 2023a)	✓	✓	✓		✓					
LEGENT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison with other embodied AI platforms. Real.: the scenes and physics in the environment are realistic. Anim.: the platform supports humanoid animation. Inter.: humans can interact with the agent directly. Lang.: the agent can perform language interaction. Access: the platform can be publicly accessed. Cross.: the environment is cross-platform and does not require specialized systems or hardware. Scene: the platform can generate various scenes automatically. Asset: the platform can utilize an unlimited variety of external assets. Style: the environment offers multiple rendering styles to provide various visual effects. Data: Whether it supports automatic generation of large-scale trajectory data for training embodied agents.

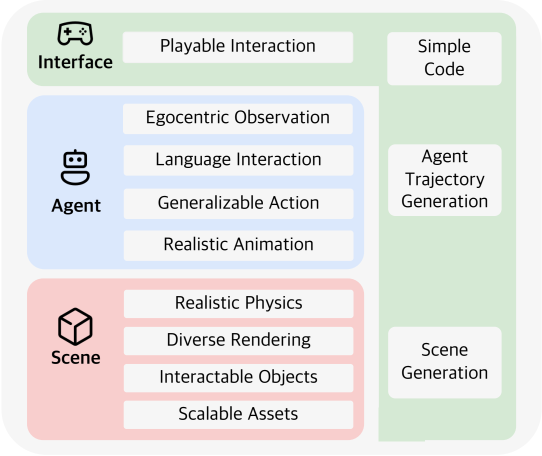


Figure 2: Features of LEGENT.

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 LLMs and 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.

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. LEGENT introduces another facet of generalization via diverse rendering. Unlike the fixed stylized renderings in games and the emphasis on photorealism in robotics, LEGENT integrates these styles by customizing the rendering functions, which allows easy transitions between rendering styles to accommodate different requirements for flexible usage. visually diverse environments

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 at runtime, 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), as illustrated in Fig. 3. We choose glTF as the import format for its openness and broad compatibility. This feature grants users the flexibility to customize the scene by strategically placing these assets or integrating them seamlessly 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

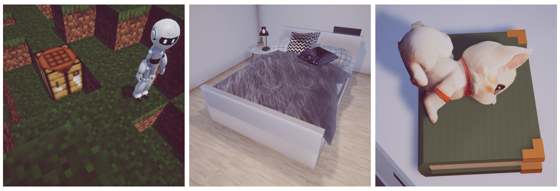


Figure 3: Examples of importing external assets: user-supplied assets (left); existing datasets (middle); assets generated by generative models (right).

Actions	Description
Speak	Send a message.
Move*	Move forward by a specified distance.
Rotate*	Adjust the view horizontally or vertically.
Interact	Grab, put, open, or close targeted objects.

Table 2: List of actions in LEGENT. * means the action is continuous (meters or degrees).

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 has the potential to connect the extensive knowledge in LLMs and 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 depends on internal states and annotations for execution, or requires an additional neural executor module compatible with a planning module (Driess et al., 2023). *Control*, on the other hand, refers to the action expression like "*move forward 1 meter, rotate to the right 30 degrees*", which is considered more generalizable. In LEGENT, we use *control*, targeting generalizing to new environments including 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 LMMs: (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 an agent moves incrementally in a stepwise manner, a process that creates less effective data for training large models. This design draws parallels to the use of keyframes in video processing and making direct training of autoregressive LMMs (Alayrac et al., 2022; Awadalla et al., 2023; Lin et al., 2024) feasible. Specifically, the actions currently supported in LEGENT are shown in Table 2. Considering the current capability of LMMs, LEGENT temporarily omits the complex control of agents' body joints. Adding these degrees of freedom to allow more flexible action will be explored in the future.

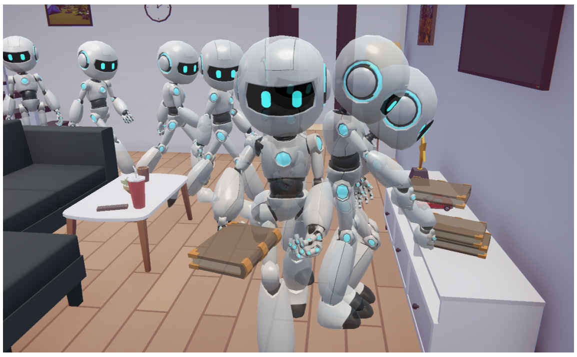


Figure 4: An example of humanoid animations, demonstrating accurate object grasping and body movement through spatial planning and inverse kinematics.

Realistic Animation. LEGENT features precise humanoid animations using inverse kinematics and spatial algorithms, enabling lifelike movements, as shown in Fig. 4. It is important for enhancing nonverbal interactions in AI systems and contributes to robotic control and text-to-motion research. Also, when combined with egocentric vision, it offers a cost-effective alternative for immersive experiences similar to Ego4D (Grauman et al., 2022), which requires a huge cost to collect.

3.3 Interface

Our platform offers a user-friendly interface for researchers to integrate LLMs and LMMs with the embodied environment easily, with little need for expertise in 3D environments. Detailed guidance is available in our documentation.

Playable Interaction. 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 and simplifies the process of conducting hands-on demonstrations.

Simple Code. LEGENT is equipped with a Python toolkit to enable the interaction between the agent and the environment. The coding interface of our Python toolkit is simple, with concise code examples available in our documentation.

Scene Generation Interface. Our platform incorporates various scene-generation techniques. Currently, we support methods including procedural generation and LLM-based generation. We provide a straightforward JSON format for specifying a scene, enabling users to easily develop their own scene generation methods.

Agent Trajectory Generation Interface. We offer an agent trajectory generation interface specifically designed for training LMMs. Using this interface, users can create training datasets that consist of egocentric visual records and corresponding ground truth actions paired with task instructions or queries, as elaborated in Section 4.3.

Hardware Requirements. LEGENT is cross-platform. It can run effortlessly on personal computers without demanding particular prerequisites or complex setups, and it facilitates connections to remote servers for training and deployment, thus enhancing its accessibility.

4 Data Generation

The second part of LEGENT is a scalable data generation pipeline. It aims at exhaustively exploiting the inherent supervision from simulated worlds and supporting large-scale training of general-purpose embodied agents. Here we elaborate on the implementation of our data generation framework.

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 generation algorithm created by 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. During the process, spatial algorithms are used to prevent object overlap and ensure precise placement. 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. In addition, instead of employing human annotators as previous work does, we utilize LLMs for asset annotation, establishing an efficient *automatic asset annotation* pipeline that facilitates future asset expansion.

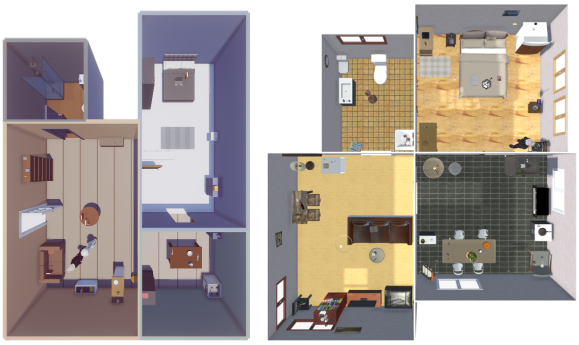


Figure 5: Examples of generated scenes.

Language Guided Generation. We implement methods in Holodeck (Yang et al., 2023b) into LEGENT and offer an LLM-powered interface to generate single or multi-room indoor scenes given any natural language query. This process resembles procedural generation but is driven by LLMs instead of human-written programs. Instead of using the depth-first-search solver in Holodeck, we ask LLMs to determine the exact locations of doors and floor objects, granting LLMs more control over the room layout. Collision detection is used to prevent interference between objects during generation.

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 for task generation.

Task Generation for Given Scenes. In this strategy, we 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 ap-

Task	Intermediate Code
come here	goto_user()
go to A	goto(a)
pick up A	goto(a) target(a) interact()
bring me A	goto(a) target(a) interact() goto_user()
where is A	find(a) speak(C)

put A on B	goto(a) target(a) interact()
	goto(b) target(b) interact()

Table 3: Currenly provided task templates and intermediate code templates. A is the object's name, and a is the object's environment identifier. C denotes the name of the receptacle on which a is placed.

proach efficiently generates large-scale samples for specific tasks based on the scene generation algorithm. 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. As shown in Table 3, we provide some basic task templates that are ideal for creating large-scale scenes, which are particularly useful for pretraining *fundamental capabilities* of embodied control, spatial comprehension, and basic language grounding across diverse scenes.

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.

We use LLMs and controllers to label the ground truth actions. 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. Each controller operates in a step-by-step manner in the environment, with visual observations collected during the process. This approach is consistent with the concept of Task and Motion Planning (TAMP) (Garrett et al., 2021) in robotics, where the LLMs and the controllers respectively fulfill the functions of task planning and motion planning.

We demonstrate this process using an example task "*Where is the orange?*". As shown in Figure 6, to finish the task, the agent needs to search the room and answer the question. LLMs map the task to the appropriate code usage, determine the object identifier of the orange in the scene, and recognize its placement from the state description, thereby generating the following intermediate code:

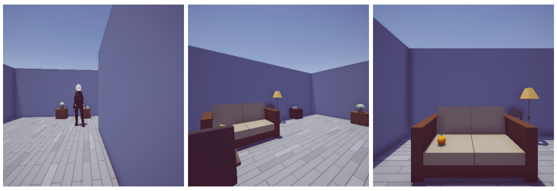


Figure 6: A generated trajectory for task "Where is the orange". The actions for the three observations are: 1. *rotate_right(-59)*; 2. *move_forward(1.2), rotate_right(-35)*; 3. *speak("It's on the sofa.")*.

```
find(36) # object identifier of orange
speak("It's on the sofa.")
```

Note that the code-writing is annotation-oriented. Even though LLMs can directly answer the question from the state description, it still invokes "*find*". Then the code "*find*" is instantiated as a multi-step controller that utilizes pathfinding algorithms (Hart et al., 1968) incorporating visibility checks. The pathfinding algorithm 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 6, the agent needs to rotate 59 degrees to the left to orient to the next waypoint, resulting in the action "*rotate_right(-59)*". Similarly, in the second observation, the agent needs to perform certain actions to move to the subsequent waypoint. This multi-step controller concludes when the target object enters the agent's field of view. LEGENT records visual observations and actions during this process as a trajectory, which can be exported as a video or an image-text interleaved sequence. The actions use a unified code representation, compatible with the outputs of LMMs.

Similar to "*find*", each intermediate code is designed with the ability to generate optimal controls using the internal world states. In addition, each task template mentioned in Section 4.2 is equipped with intermediate code templates, as shown in Table 3, eliminating the need for LLMs in large-scale data generation for specific tasks.

4.4 Prototype Experiments

We conduct a prototype experiment to assess the utility of generated data on two embodied tasks: "Come Here" for navigation and "Where Is" for embodied question answering (Das et al., 2018). Task

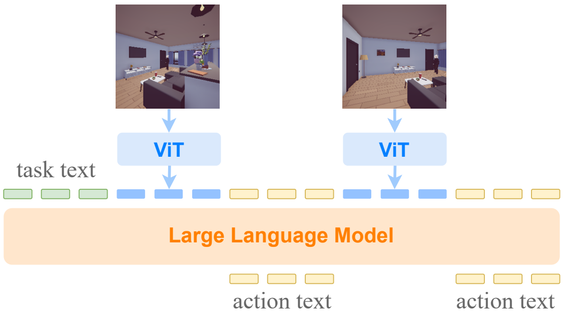


Figure 7: The vision-language-action(VLA) model architecture used in the prototype experiments.

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 4: Success rates on two embodied tasks. *VILA-Sep* denotes models fine-tuned separately for each task, whereas *VILA-Joint* refers to models trained jointly on both tasks. * means generalization test.

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., 2024) as our backbone due to its capability in interleaved inputs. We train the vision-language-action model to predict current action based on task descriptions and interleaved context of previous observations and actions, as illustrated in Fig. 7.

The results presented in Table 4 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.

4.5 Demo of LEGENT

The demo video of LEGENT is available at the link[1], which is partially shown in Fig. 1. The demonstration exemplifies the engagement with embodied agents in LEGENT, primarily leveraging LLMs and controllers described in Section 4.3. With advancements in LMMs' capability of ego-centric perception and control, we foresee the evolution of this demonstration into a fully embodied experience, independent of any extra internal information. We will also pursue this goal by further scaling the data generation for model training.

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. By bridging the gap between embodied AI and LMM's development, we hope LEGENT inspires research in this field. We are committed to the ongoing development of LEGENT, making it more scalable and user-friendly. 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 to make actions more applicable to the real world. (4) Improving scene generation and integrating text-to-3D and image-to-3D methods to support more diverse and realistic scenes.

Limitations

In the context of imitation learning, the persistent challenge of insufficient exploration and handling out-of-distribution inputs during inference underscores the need for further enhancements and strategies within the data generation pipeline, a component that is currently not integrated into our system. Furthermore, large-scale experiments have yet to be conducted. We leave this to the future work.

Ethics Statement

The development of LEGENT prioritizes ethical considerations across all aspects of its use and implementation. We uphold data privacy and security, ensuring compliance with relevant data protection

[1] https://video.legent.ai

laws. We strictly adhere to legal standards and encourage ethical use of our platform. We are committed to continuous evaluation of the ethical implications of our work and engaging with the community to address emerging concerns and ensure a positive impact on society.

Acknowledgements

This work is supported by the National Key R&D Program of China (2020AAA0106502), the National Natural Science Foundation of China (No. 62236011, 623B2065) and a grant from the Guoqiang Institute, Tsinghua University.

References

Josh Abramson, Arun Ahuja, Iain Barr, Arthur Brussee, Federico Carnevale, Mary Cassin, Rachita Chhaparia, Stephen Clark, Bogdan Damoc, Andrew Dudzik, et al. 2020. Imitating interactive intelligence. *arXiv preprint arXiv:2012.05672*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. 2023. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*.

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

Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. 2016. Deepmind lab. *arXiv preprint arXiv:1612.03801*.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. 2024. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*.

Qiuyu Chen, Marius Memmel, Alex Fang, Aaron Walsman, Dieter Fox, and Abhishek Gupta. 2023. Urdformer: Constructing interactive realistic scenes from real images via simulation and generative modeling. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*.

Murtaza Dalal, Ajay Mandlekar, Caelan Garrett, Ankur Handa, Ruslan Salakhutdinov, and Dieter Fox. 2023. Imitating task and motion planning with visuomotor transformers. *arXiv preprint arXiv:2305.16309*.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10.

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

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. 2022. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2024. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36.

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

Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. 2021. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4:265–293.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. 2018. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4089–4098.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012.

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

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

Jinyi Hu, Yuan Yao, Chongyi Wang, SHAN WANG, Yinxu Pan, Qianyu Chen, Tianyu Yu, Hanghao Wu, Yue Zhao, Haoye Zhang, Xu Han, Yankai Lin, Jiao Xue, dahai li, Zhiyuan Liu, and Maosong Sun. 2024. Large multilingual models pivot zero-shot multimodal learning across languages. In *The Twelfth International Conference on Learning Representations*.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. 2023. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*.

Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. 2016. The malmo platform for artificial intelligence experimentation. In *Ijcai*, volume 16, pages 4246–4247.

Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. 2023. 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*, pages 10813–10823.

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

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

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2023. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024. Vila: On pretraining for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv: 2308.03688*.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. 2021. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*.

Junhyuk Oh, Valliappa Chockalingam, Honglak Lee, et al. 2016. Control of memory, active perception, and action in minecraft. In *International conference on machine learning*, pages 2790–2799. PMLR.

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

Xavi Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Ruslan Partsey, Jimmy Yang, Ruta Desai, Alexander William Clegg, Michal Hlavac, Tiffany

Min, Theo Gervet, Vladimir Vondrus, Vincent-Pierre Berges, John Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. 2023a. Habitat 3.0: A co-habitat for humans, avatars and robots.

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

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. 2023b. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. 2022. A generalist agent. *arXiv preprint arXiv:2205.06175*.

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. 2023. Meshgpt: Generating triangle meshes with decoder-only transformers. *arXiv preprint arXiv:2311.15475*.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. 2023. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE.

DeepMind Interactive Agents Team, Josh Abramson, Arun Ahuja, Arthur Brussee, Federico Carnevale, Mary Cassin, Felix Fischer, Petko Georgiev, Alex Goldin, Mansi Gupta, et al. 2021. Creating multimodal interactive agents with imitation and self-supervised learning. *arXiv preprint arXiv:2112.03763*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. 2023. Chatgpt for robotics: Design principles and model abilities. *arXiv preprint arXiv:2306.17582*.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. Scienceworld: Is your agent smarter than a 5th grader? *arXiv preprint arXiv:2203.07540*.

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

Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. 2024. Crm: Single image to 3d textured mesh with convolutional reconstruction model. *arXiv preprint arXiv:2403.05034*.

Zehao Wen, Zichen Liu, Srinath Sridhar, and Rao Fu. 2023. Anyhome: Open-vocabulary generation of structured and textured 3d homes. *arXiv preprint arXiv:2312.06644*.

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

Zhou Xian, Theophile Gervet, Zhenjia Xu, Yi-Ling Qiao, Tsun-Hsuan Wang, and Yian Wang. 2023. Towards generalist robots: A promising paradigm via generative simulation. *arXiv preprint arXiv:2305.10455*.

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

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. 2023a. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. 2023b. Holodeck: Language guided generation of 3d embodied ai environments. *arXiv preprint arXiv:2312.09067*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John Turner, et al. 2023. Homerobot: Open-vocabulary mobile manipulation. *arXiv preprint arXiv:2306.11565*.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. 2020. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR.

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. 2023. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*.