Computer-Use Agents Are Embodied Generalists

Pranjal Aggarwal **Sean Welleck**
Carnegie Mellon University

Abstract

In many areas of AI, general-purpose models have increasingly replaced handcrafted methods and specialized architectures. We argue that embodied AI is poised for a similar shift. Our key claim is that embodied tasks can be formulated as interaction with digital software. Specifically, we build software that renders an agent's view of a physical environment on a screen and provides controls for acting in it. A computer-use agent (CUA) can then perform embodied tasks through the same screenshots and keyboard and mouse actions that it uses to operate ordinary software. Using a single CUA design and changing only the software, our entries win first-, second-, third-, and sixth-place prizes across four ECCV workshop competitions, without any competition-specific agent tuning. These results suggest that the skills CUAs learn on software, such as visual grounding, precise clicking, and exploration, already carry over to embodied tasks, giving CUAs a far broader scope in embodied AI than previously realized. We therefore advocate that computer-use agents can be embodied generalists. Further, we advocate that, by following the training strategy that made computer-use agents general, embodied AI can converge on a single general model as well, ushering in a new era of general embodied AI.[1]

1 Introduction

In many areas of AI, general-purpose models have increasingly replaced handcrafted methods and specialized architectures (Sutton, 2019; OpenAI et al., 2025; Luong and Lockhart, 2025). We argue that embodied AI, which studies agents that perceive and act in physical environments, is poised for a similar shift (Open X-Embodiment Collaboration et al., 2024; Gemini Robotics Team et al., 2025). In particular, we explore connecting embodied AI with computer-use agents: general-purpose digital agents that operate diverse software applications through screenshots, mouse clicks, and keyboard inputs. Although operating a computer and acting in a physical environment may seem quite different, we observe that both involve similar challenges, including visual grounding, exploration, and adapting actions based on feedback from the environment (Yang et al., 2025a; Zhang et al., 2025a; Huang et al., 2022).

Our key insight is that embodied tasks can be formulated as interaction with digital software. Specifically, we construct software that renders observations on a screen and provides controls for acting in the physical environment (Figure 1). A computer-use agent can then observe the environment, take actions, and inspect their consequences using the same interface that it uses to operate ordinary desktop software. Task-specific functionality resides in the software rather than the agent design, allowing us to use existing computer-use agents across different embodied tasks without developing specialized models.

To investigate our approach, we develop software applications for four embodied AI tasks: vision-language navigation, dialogue-driven navigation, language-guided 3D segmentation, and referring video segmentation. We use a single, minimal computer-use agent harness across all four applications, without competition-specific agent tuning, and enter 4 ECCV 2026 workshop competitions. Our

[1]Website: `https://pranjal2041.github.io/cua-embodied`

Preprint.

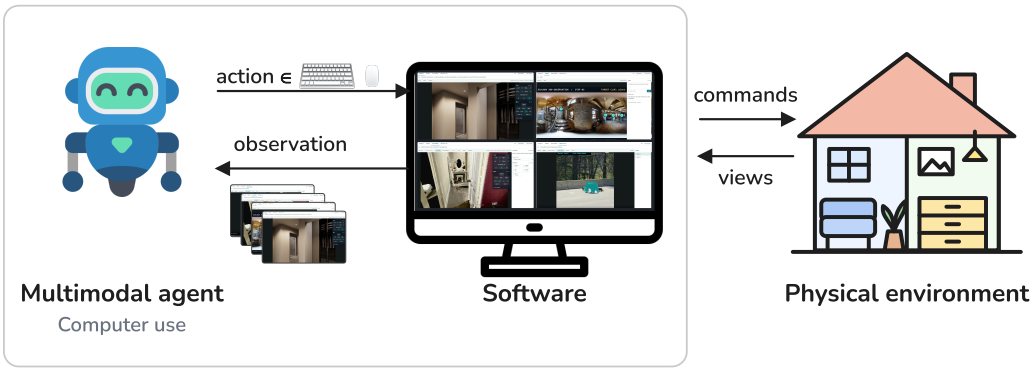


Figure 1: Computer-use agents can operate any software through keyboard and mouse actions while observing the screen. We argue that physical environments are no different. Once exposed through software, they become just another interface for the same agent.

entries place first in OpenSUN3D, second in MeViSv2-Text, third in DialNav, and sixth in the VLNVerse competitions. These results demonstrate that computer-use agents can achieve strong performance across diverse embodied tasks by interacting with task-specific software through a shared agent design.

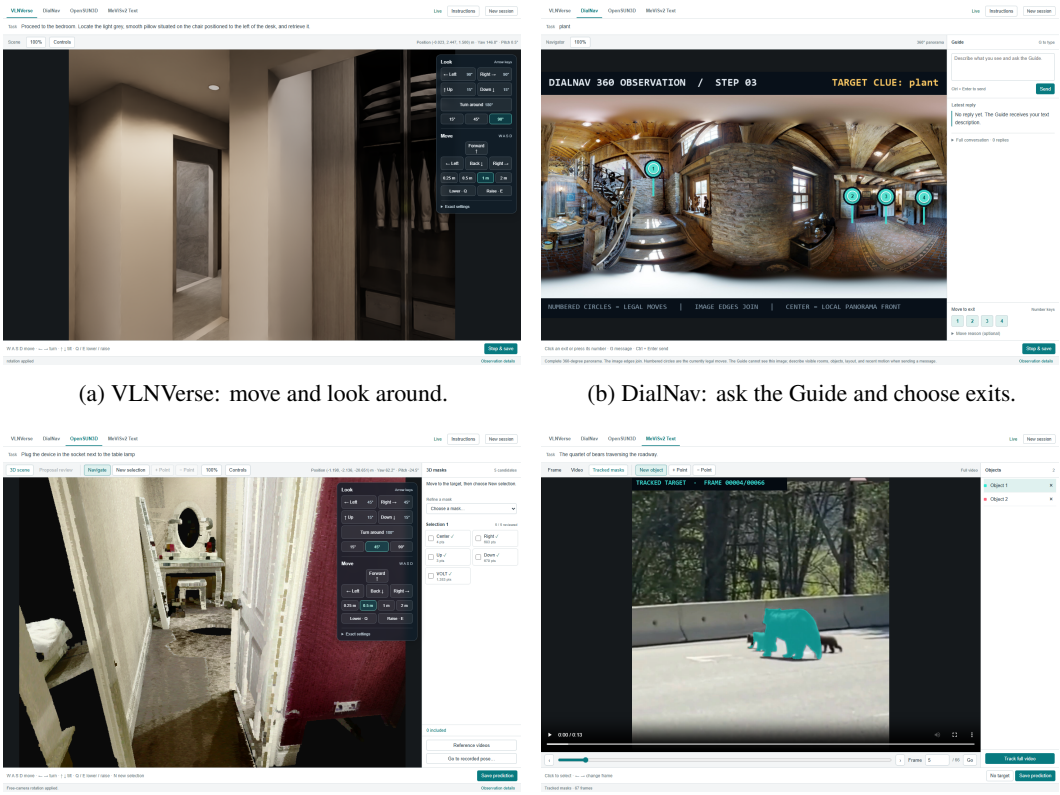
Beyond these results, we argue that embodied AI can follow the same path that computer use has already taken. Only two years ago, computer-use agents faced the challenges that embodied AI faces today. They struggled to click precisely and relied on overlays or separate grounding models to locate targets (Yang et al., 2023; Zheng et al., 2024; Gou et al., 2025), much as most VLAs today add a dedicated action head for continuous control (Physical Intelligence et al., 2025; NVIDIA et al., 2025). They also lagged far behind agents with task-specific APIs (Xie et al., 2024; Aggarwal and Welleck, 2026c; Song et al., 2025). Today, the same model that writes code and solves mathematics problems can also operate computers by writing click coordinates as ordinary text (Anthropic, 2024; Quiros, 2026), and the strongest agents perform comparably whether they act through the screen or through APIs (NeoCognition, 2026). This progress came from training for precise visual grounding, building environments for reinforcement learning, and making interaction fast enough for real-time use (Gou et al., 2025; Aggarwal et al., 2026b; Wang et al., 2026; Aggarwal et al., 2026a). Each of these has a direct counterpart in embodied AI. Simulators can generate grounding data at scale (Yuan et al., 2024), embodied environments can be used for reinforcement learning (Li et al., 2026), and faster inference can enable real-time control (Black et al., 2025b). Since our approach exposes embodied environments through software, advances in computer-use agents can also transfer to embodied AI directly, and *embodied tasks can become part of the training of future general-purpose agents*. We therefore believe that embodied AI will converge on a single general model, just as computer use did (OpenAI, 2026c; Anthropic, 2026).

2 Methodology

Our goal is to provide sparks of evidence for our hypothesis that the capabilities computer-use agents learn on digital software also generalize to embodied environments. We entered four different ECCV workshop competitions, and evaluated a single computer-use agent on their official test sets. We first describe the computer-use agent shared across all four, and then the software we built for each competition.

2.1 A shared computer-use agent

We keep the computer-use agent's design fixed and construct software for each competition. The agent receives the task description, the software's instructions, and any reference images or videos. Let q denote these task inputs, $\mathcal{S}_k$ the software for competition k, and s_t its state at step t. The software presents a visual observation o_t, consisting of a screenshot and any video returned by an operation. Given the interaction history $h_t = (o_0, a_0, \ldots, o_t)$, the agent π chooses a keyboard or mouse action



(a) VLNVerse: move and look around.

(b) DialNav: ask the Guide and choose exits.

(c) OpenSUN3D: explore and select 3D regions.

(d) MeViSv2: select objects and review tracks.

Figure 2: The shared computer-use agent operates four software applications through keyboard and mouse actions. Each application provides task-specific controls and displays the outcomes.

a_t:

$$a_t \sim \pi(\cdot \mid q, h_t), \qquad (s_{t+1}, o_{t+1}) = \mathcal{S}_k(s_t, a_t). \tag{1}$$

The software executes the action and returns the updated observation. The agent repeats this loop until it presses submit through the interface, at which point the software exports the trajectory or prediction.

We use Gemini 3.5 Flash and Gemini 3.6 Flash with the same agent implementation, system prompt, and keyboard and mouse action space across the four software applications. Each application provides controls for its internal operations, including movement, dialogue, segmentation, and mask propagation. The agent chooses when to use these operations, interprets their outputs, and decides what to do next. It can issue individual inputs or batches of up to 16 inputs; each operation finishes before the next begins, and a screenshot follows the batch. We retain the interaction text and remove the oldest ten screenshots when the context reaches twenty. Complete reference videos and generated overlay videos remain in context. Figure 2 shows the four interfaces.

2.2 VLNVerse: navigation through a 3D scene

VLNVerse (Lin et al., 2025) requires navigating to a destination described in language. We construct a 3D scene viewer with a first-person image and controls for translation and rotation. The agent moves forward, backward, sideways, or vertically using keyboard shortcuts or on-screen buttons. It can turn, look up or down, and adjust the movement distance and rotation angle. Each operation updates the camera pose and renders a new image. The agent uses these observations to identify landmarks, explore rooms, and approach the target. The same viewer supports the coarse and fine tracks. Selecting the stop control saves the trajectory and final pose.

2.3 DialNav: navigation with a conversational guide

DialNav (Han et al., 2025) requires reaching a destination through dialogue with a Guide. Our software displays a 360° panorama, numbered exits to adjacent viewpoints, a target clue, and a message panel. The agent types descriptions and questions, reads the Guide's replies, and clicks exits to move. A Guide model within the software receives the destination, a scene atlas, and route previews in a separate context. It infers the agent's location from its descriptions and returns natural-language directions. The agent uses these directions and the panorama to choose each movement. The stop control saves the trajectory and dialogue.

2.4 OpenSUN3D: language-guided 3D segmentation

The OpenSUN3D challenge (OpenSUN3D, 2026) uses SceneFun3D (Delitzas et al., 2024) tasks that describe an action and require a 3D mask of the corresponding interaction region, such as a handle for opening a drawer. Our software renders the colored Faro point cloud and provides free-camera movement, rotation, and point selection. Reference videos supply scene context and shortcuts to recorded camera poses. The agent explores the 3D scene, approaches the target, and clicks the relevant component. The renderer maps this click to a visible Faro point. Within the software, PinPoint3D (Zhang et al., 2025b) produces point-prompted masks, while a VOLT model fine-tuned on SceneFun3D supplies affordance-based proposals. The interface displays each candidate from four viewpoints. The agent can refine a selection with positive or negative points, move to another viewpoint, or make a new selection. It then chooses which masks to save, retaining their original Faro point indices.

2.5 MeViSv2: referring video segmentation

The MeViSv2 (Ding et al., 2025) text track requires segmenting the objects described by a motion expression throughout a video. Our software provides a video player, a frame selector, and controls for adding, refining, and removing objects. The agent receives the complete video and expression, chooses a frame, and clicks each target. SAM3 (Carion et al., 2025) segments these objects and propagates their masks through the video. The agent receives the complete tracked-mask overlay and can inspect individual frames. It can add positive or negative points, remove an object, or select another object before running propagation again. The save control exports the masks of all remaining objects. A no-target control exports an empty mask sequence.

3 Results

3.1 Competition results

The same computer-use agent, with only its software changed, won prizes in all four competitions, spanning navigation, dialogue-driven navigation, 3D segmentation, and video segmentation. It placed first in OpenSUN3D, second in MeViSv2-Text, third in DialNav, and sixth in VLNVerse. We report the official test scores of our competition entries, together with the best other entry in each competition, in Table 1. In OpenSUN3D, the agent finished ahead of every other entry by more than 4 points in AP50 and 10 points in AP25. There, it explored a 3D scan and clicked the exact part to interact with, such as a drawer handle, much as it clicks a small button on a screen. In VLNVerse, the agent placed sixth on success but followed the reference routes more closely than all but two of the 74 entries, with the third-highest NDTW on both tracks. In MeViSv2-Text, it placed second ahead of an entry with a higher $\mathcal{J}\&\mathcal{F}$, because it more often recognized when no object matched the expression (88.89 versus 83.33 no-target accuracy). Each of these results rests on abilities the agent already uses on digital software, namely locating targets, pointing at them precisely, and exploring its environment.

VLNVerse's SR@3 measures the fraction of episodes ending within three metres of the target. Avg SR@3 averages it over the coarse and fine tracks, and NDTW measures how closely a path follows the reference route. DialNav reports the endpoint success rate and success weighted by path length (SPL). OpenSUN3D evaluates 3D instance masks using average precision at 50% and 25% intersection over union. MeViSv2's Final score averages $\mathcal{J}\&\mathcal{F}$ with the accuracy of predicting target presence and absence.

Table 1: Official test scores of our competition entries and the best other entry for each metric, taken from the official challenge results (EMR, 2026; EAD, 2026; OpenSUN3D, 2026; Liu et al., 2026). All scores are expressed on a 0–100 scale.

Competition	Place	Metric	Ours	Best other
VLNVerse	6th	Avg SR@3	87.30	90.70
		NDTW (coarse / fine)	95.0 / 96.8	96.3 / 97.6
DialNav	3rd	Endpoint success rate	65.26	–
		SPL	52.87	–
OpenSUN3D	1st	AP50	25.88	21.45
		AP25	36.52	26.50
MeViSv2-Text	2nd	Final	86.72	90.81
		$\mathcal{J}\&\mathcal{F}$	74.96	79.22

3.2 GUI operations or text APIs?

A natural question is whether we need to construct GUIs or could instead provide direct text APIs for navigation and other operations. We view computer use as operating a digital interface. GUIs and APIs are two choices for that interface. We expect sufficiently capable computer-use agents to achieve the same task performance with either interface, although efficiency may differ.

We evaluate this by comparing GUI and API interaction on a subset of tasks from each of the four competitions. Each pair uses the same model, task inputs, and software operations. We compare successive generations of Gemini Flash to examine how improvements in computer use affect the performance gap between interfaces.

Table 2 shows the results on the validation sets. The two interfaces performed within four points of each other, and the GUI scored higher in two of the four competitions.

Table 2: GUI and API interaction on the validation set of each competition, with the same model, task inputs, and software operations. All scores are expressed on a 0–100 scale.

Competition	Metric	API	GUI	GUI − API
VLNVerse	Avg SR@3	83.9	85.2	+1.3
DialNav	Endpoint success rate	67.4	63.4	−4.0
OpenSUN3D	AP50	26.7	28.1	+1.4
MeViSv2-Text	Final	95.8	94.6	−1.2

This expectation is consistent with prior work. The performance gap between GUI-only and API-assisted interaction has shown to be narrowing across claude generations (Aggarwal and Welleck, 2026c), while more recently, the strongest evaluated agents achieve comparable or higher task success through GUIs than through APIs (NeoCognition, 2026).

4 Position: Computer-use agents are embodied generalists

Claim 1: Representing physical interfaces in software lets us apply the methods that improved computer-use agents to embodied AI

The challenges facing embodied AI/robotics today are similar to those faced by computer-use agents two years ago. Both require an agent to ground fine-grained actions in visual observations and adapt as its environment changes (Cheng et al., 2024; Yuan et al., 2024; Lee et al., 2026; Yang et al., 2025a) (Figure 3). For computer use, this means locating a target on a screen and clicking at the correct coordinates. For robotics, it means locating targets in three dimensions and coordinating motor commands to achieve the intended positions and orientations. Representing physical interfaces in software or simulations lets us apply the same methods for training agents to perform these interactions. Our results in Section 3 offer initial evidence: using the same agent and changing only the software, we performed competitively in four embodied competitions.

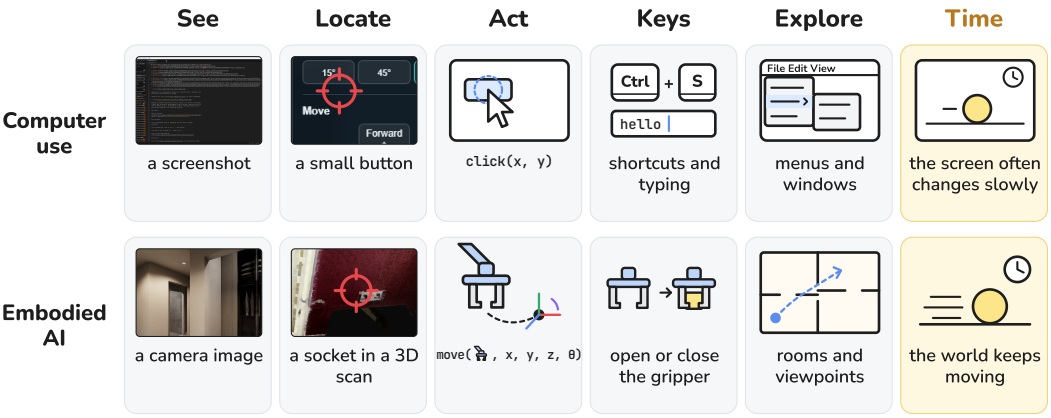


Figure 3: Computer use and embodied AI require much of the same from an agent. In both, it must see its surroundings, locate a target, act on it precisely, issue discrete commands, and explore. The main difference is time, as the screen often changes slowly while the physical world keeps moving.

Fine-grained motor control and real-time physical interaction may seem out of reach for computer-use agents. Yet only two years ago, precise clicking seemed equally out of reach: computer-use agents struggled to operate digital interfaces (Xie et al., 2024) and lagged behind agents equipped with task-specific APIs (Aggarwal and Welleck, 2026c; Song et al., 2025). As discussed in the previous section, the strongest agents now achieve comparable task performance through either interface (NeoCognition, 2026). We therefore argue for following the same development strategies in embodied AI, from learning precise, visually grounded actions to improving reliability and speed.

Claim 2: Embodied AI will converge on a single general model, as computer use did

Much of embodied AI today is built on vision-language-action models (VLAs), a special case of vision-language models further fine-tuned for robotic control (Brohan et al., 2023; Kim et al., 2024; Black et al., 2025a; Kawaharazuka et al., 2025). Since robot actions are continuous, and discrete tokens seem too coarse to express them precisely, most VLAs also add a dedicated action head, typically trained with diffusion or flow matching (Black et al., 2025a; NVIDIA et al., 2025; Li et al., 2024; Kawaharazuka et al., 2025). But computer-use agents faced exactly this problem only two years ago. A screen of a thousand by a thousand pixels is effectively a continuous space, and early agents could not act in it precisely on their own, relying instead on Set-of-Mark overlays (Yang et al., 2023; Xie et al., 2024) or separate grounding models (Gou et al., 2025). Yet with enough training, neither turned out to be necessary: today, a single model operates computers by simply writing click coordinates as ordinary text tokens, without any special head or tokenizer (Cheng et al., 2024; Qin et al., 2025; Bai et al., 2025).

We therefore believe that embodied AI will follow the same path, and that specialized action heads will go the way of specialized grounding models (Goyal et al., 2025; Hancock et al., 2026). In fact, RT-2 and OpenVLA already show that robot actions can be expressed as tokens in the language model's own vocabulary, without any action head (Brohan et al., 2023; Kim et al., 2024). To support this claim, we perform a preliminary investigation in Gym-Anything for Robotics (Aggarwal and Welleck, 2026a). We train $\pi_{0.5}$ (Physical Intelligence et al., 2025) and a standard vision-language model (Qwen3.5) with 9B parameters on the same 1,000 demonstrations for two tasks in a gift box simulation with two robot arms. *We add no action head and do not even change the tokenizer.* The only change is the format in which actions are written as text, similar to computer-use agents. Yet this plain vision-language model matches $\pi_{0.5}$, succeeding in 56% of attempts, against 59% for $\pi_{0.5}$ across 1000 unseen instances. The model is also fast. As with $\pi_{0.5}$, the model receives a new camera image every second and either writes an action, which can span a longer duration, or chooses not to act. The 9B model takes about 400 milliseconds on average to generate an action on a single L40S GPU. Because each action can span a longer duration, decoding takes up only about 20% of the total time and prefill about 10%, with the remaining 70% idle. However, this setting has only two robot arms. More complex robots, such as humanoids, have many more motors, and writing every state and action for them as plain text would make each action slower to generate. Such robots still need

a more efficient format for states and actions, and possibly a better tokenizer (Pertsch et al., 2025; Kawaharazuka et al., 2025). And even if we use additional action heads, or whatever that format turns out to be, the most important thing is, embodied AI will end up exactly where computer use did, with a single model that writes code, solves mathematics problems, operates computers, and controls robots (Yang et al., 2025a; Reed et al., 2022; Gemini Robotics Team et al., 2025).

Claim 3: The path to embodied generalists follows the same playbook as computer-use agents

What enabled the progress of computer-use agents over the last two years? We view the development of computer-use agents in three phases, each with a direct counterpart in embodied AI.

The first phase was visual grounding. Digital interfaces provide a way to generate supervision automatically: the underlying software can expose the identities and coordinates of interface elements, which can then be paired with screenshots. This approach has been used to generate grounding data at scale (Gou et al., 2025). Simulators offer the same opportunity for embodied AI: they can expose the identities and 3D positions of objects, which can be paired with rendered views to generate grounding data at scale (Yuan et al., 2024). Of course, this requires simulations with high enough visual fidelity, an area where coding models are clearly and continuously improving.

The second phase was building environments for reinforcement learning, following concurrent approaches used in other domains such as coding (Pan et al., 2025; Yang et al., 2025b). Automated pipelines made it possible to generate tasks, initialize environments, and verify outcomes at scale (Aggarwal et al., 2026b; Wang et al., 2026). These environments then let agents learn through reinforcement learning at scale. For embodied AI, this means creating such environments in the form of simulations with realistic physics, with tasks and verifiers available through software. We further argue that embodied environments have an even stronger asymmetry between task creation and verification Aggarwal and Welleck (2026a), which can be used to provide more accurate rewards or to increase task difficulty over time. We also note that reinforcement learning benefits from a strong initial policy, and training such a policy requires trajectories for pretraining or midtraining. We are therefore not advocating against teleoperation, whether in the real world or in simulation, or against egocentric videos, although this data may matter mainly in pretraining and midtraining, as in the training recipes of existing LLMs. How much of this data is needed also remains unclear, but based on preliminary results presented in this paper, we believe that perhaps much of it will not be needed, and that research on environments for reinforcement learning will remain more important.

The third phase is speed and real-time interaction, which is especially important for robotics. An agent must choose and execute actions quickly enough to keep up with changes in its environment. This requires optimizing completion time alongside task success (Aggarwal et al., 2026a). Even without such optimization, the 9B model in Claim 2 took about 400 milliseconds per action and was idle for 70% of the time. We can pursue this through rewards for completing tasks successfully with less reasoning or in less time (Aggarwal and Welleck, 2025; Arora and Zanette, 2025), faster inference hardware (Lie, 2023; Abts et al., 2020; He et al., 2025), and harnesses that interleave quick actions with slower planning (Shi et al., 2025; NVIDIA et al., 2025). Such agentic harnesses organize how an agent plans, acts, and checks its progress (OpenAI, 2026a; Koh et al., 2026), and embodied tasks can reuse them once they are exposed through software.

The sequence is therefore the same: learn grounding, build environments for reinforcement learning, and optimize for real-time interaction.

So what should be next steps for Embodied AI?

We therefore advocate for 3 research directions we believe would become fundamental for embodied AI research: a.) improving visual grounding through data generated automatically from simulators (Pfaff et al., 2026; Yuan et al., 2024), b.) building robotic environments as simulations for robotics, exposed through software that computer-use agents can operate and be RL trained on (Aggarwal and Welleck, 2026a; Wang et al., 2024; Hua et al., 2024), and c.) optimizing agents for real-time interaction (Aggarwal and Welleck, 2026b; Tang et al., 2026b) through architecture research (Choudhury et al., 2024; Tang et al., 2026a; Aggarwal and Welleck, 2026d) and by rewarding speed alongside task success (Arora and Zanette, 2025; Aggarwal and Welleck, 2025). These directions call on embodied AI researchers to expose their simulators and robots through software, and on developers of computer-use agents to include such software in training.

5 Conclusion

In this paper, we argued that computer-use agents can already act as embodied generalists, because embodied tasks can be formulated as interaction with digital software. Using a single computer-use agent and changing only the software, we won prizes in four ECCV workshop competitions, from navigation to video segmentation. Operating that software through a GUI cost little compared to a text API, and a vision-language model with no action head matched $\pi_{0.5}$ when trained on the same robot demonstrations. We then traced how computer-use agents became general and superhuman over the last two years, through visual grounding, environments for reinforcement learning, and speed, and argued that embodied AI should follow the same playbook. We believe embodied AI is headed the same way, and that its generalist version will be the same model that already writes code, solves mathematics problems, and operates digital computers.

Acknowledgements

We thank Daytona for their generous support in sandbox compute credits, and Google for their generous support through a Gemini API research grant. We thank the organizers of the VLNVerse, DialNav, OpenSUN3D, and MeViSv2 competitions and their ECCV 2026 workshops. All opinions expressed in this paper are the authors' own and do not necessarily reflect the views of our sponsors or the competition organizers.

References

Dennis Abts, Jonathan Ross, Jonathan Sparling, Mark Wong-VanHaren, Max Baker, Tom Hawkins, Andrew Bell, John Thompson, Temesghen Kahsai, Garrin Kimmell, Jennifer Hwang, Rebekah Leslie-Hurd, Michael Bye, E. R. Creswick, Matthew Boyd, Mahitha Venigalla, Evan Laforge, Jon Purdy, Purushotham Kamath, Dinesh Maheshwari, Michael Beidler, Geert Rosseel, Omar Ahmad, Gleb Gagarin, Richard Czekalski, Ashay Rane, Sahil Parmar, Jeff Werner, Jim Sproch, Adrian Macias, and Brian Kurtz. Think fast: A tensor streaming processor (TSP) for accelerating deep learning workloads. In *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, pp. 145–158, 2020.

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *Conference on Language Modeling (COLM)*, 2025.

Pranjal Aggarwal and Sean Welleck. Gym-Anything for Robotics: Scaling physical simulations with digital agents, 2026. URL https://arxiv.org/abs/2610.40565.

Pranjal Aggarwal and Sean Welleck. Gym-Anything-RT: Scaling real-time environments for vision-language agents, 2026. URL https://arxiv.org/abs/2610.37567.

Pranjal Aggarwal and Sean Welleck. Programming with Pixels: Can computer-use agents do software engineering? In *International Conference on Learning Representations (ICLR)*, 2026.

Pranjal Aggarwal and Sean Welleck. Visual token routing: Learning to keep tokens that matter, 2026. URL https://arxiv.org/abs/2611.06854.

Pranjal Aggarwal, Lawrence Keunho Jang, Sean Welleck, Daniel Fried, Ruslan Salakhutdinov, and Jing Yu Koh. cua-speedrun: Standardized benchmarking of the speed of computer-use agents, 2026a. URL https://arxiv.org/abs/2609.40284.

Pranjal Aggarwal, Graham Neubig, and Sean Welleck. Gym-Anything: Turn any software into an agent environment, 2026b. URL https://arxiv.org/abs/2604.06126.

Anthropic. Introducing computer use, a new Claude 3.5 Sonnet, and Claude 3.5 Haiku, 2024. URL https://www.anthropic.com/news/3-5-models-and-computer-use.

Anthropic. Introducing Claude Opus 5.5, 2026. URL https://www.anthropic.com/claude-opus-5-5.

Daman Arora and Andrea Zanette. Training language models to reason efficiently. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

Ellis Brown, Arijit Ray, Ranjay Krishna, Ross Girshick, Rob Fergus, and Saining Xie. SIMS-V: Simulated instruction-tuning for spatial video understanding, 2025. URL https://arxiv.org/abs/2511.04668.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 9313–9332, 2024.

Rohan Choudhury, Guanglei Zhu, Sihan Liu, Koichiro Niinuma, Kris M. Kitani, and László Jeni. Don't look twice: Faster video transformers with run-length tokenization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Alexandros Delitzas, Ayca Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. SceneFun3D: Fine-grained functionality and affordance understanding in 3D scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14531–14542, 2024.

Henghui Ding, Chang Liu, Shuting He, Kaining Ying, Xudong Jiang, Chen Change Loy, and Yu-Gang Jiang. MeViS: A multi-modal dataset for referring motion expression video segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):11400–11416, 2025.

EAD. Embodied Agent and Dialog Workshop, in conjunction with ECCV 2026, 2026. URL https://ead-workshop.github.io/.

EMR. Workshop on Embodied Multimodal Reasoning in Physical Environments, in conjunction with ECCV 2026, 2026. URL https://emr-workshop.github.io/.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, Jose Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, David D'Ambrosio, Sudeep Dasari, Todor Davchev, Coline Devin, Norman Di Palo, Tianli Ding, Adil Dostmohamed, Danny Driess, Yilun Du, Debidatta Dwibedi, Michael Elabd, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Leonard Hasenclever, Nicolas Heess, Brandon Hernaez, Alexander Herzog, R. Alex Hofer, Jan Humplik, Atil Iscen, Mithun George Jacob, Deepali Jain, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, Chase Kew, Jerad Kirkland, Sean Kirmani, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Jacky Liang, Yixin Lin, Sharath Maddineni, Anirudha Majumdar, Assaf Hurwitz Michaely, Robert Moreno, Michael Neunert, Francesco Nori, Carolina Parada, Emilio Parisotto, Peter Pastor, Acorn Pooley, Kanishka Rao, Krista Reymann, Dorsa Sadigh, Stefano Saliceti, Pannag Sanketi, Pierre Sermanet, Dhruv Shah, Mohit Sharma, Kathryn Shea, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Jost Tobias Springenberg, Rachel Sterneck, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Vincent Vanhoucke, Jake Varley, Grace Vesom, Giulia Vezzani, Oriol Vinyals, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Fei Xia, Ted Xiao, Annie Xie, Jinyu Xie, Peng Xu, Sichun Xu, Ying Xu, Zhuo Xu, Yuxiang Yang, Rui Yao, Sergey Yaroshenko, Wenhao Yu, Wentao Yuan, Jingwei Zhang, Tingnan Zhang, Allan Zhou, and Yuxiang Zhou. Gemini Robotics: Bringing AI into the physical world, 2025. URL `https://arxiv.org/abs/2503.20020`.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *International Conference on Learning Representations (ICLR)*, 2025.

Ankit Goyal, Hugo Hadfield, Xuning Yang, Valts Blukis, and Fabio Ramos. VLA-0: Building state-of-the-art VLAs with zero modification, 2025. URL `https://arxiv.org/abs/2510.13054`.

Leekyeung Han, Hyunji Min, Gyeom Hwangbo, Jonghyun Choi, and Paul Hongsuck Seo. DialNav: Multi-turn dialog navigation with a remote guide. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8514–8523, 2025.

Asher J. Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning VLMs into VLAs without catastrophic forgetting. In *International Conference on Learning Representations (ICLR)*, 2026.

Congjie He, Yeqi Huang, Pei Mu, Ziming Miao, Jilong Xue, Lingxiao Ma, Fan Yang, and Luo Mai. WaferLLM: Large language model inference at wafer scale. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 257–273, 2025.

Pu Hua, Minghuan Liu, Annabella Macaluso, Yunfeng Lin, Weinan Zhang, Huazhe Xu, and Lirui Wang. GenSim2: Scaling robot data generation with multi-modal and reasoning LLMs. In *Conference on Robot Learning (CoRL)*, 2024.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner Monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning (CoRL)*, 2022.

Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 13:162467–162504, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Multi-agent computer use, 2026. URL `https://arxiv.org/abs/2606.01533`.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct: Action reasoning models that can reason in space. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2585–2592, 2026.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation, 2024. URL https://arxiv.org/abs/2411.19650.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, Dehui Wang, Dingxiang Luo, Yuchen Fan, Youbang Sun, Jia Zeng, Jiangmiao Pang, Shanghang Zhang, Yu Wang, Yao Mu, Bowen Zhou, and Ning Ding. SimpleVLA-RL: Scaling VLA training via reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Sean Lie. Cerebras architecture deep dive: First look inside the hardware/software co-design for deep learning. *IEEE Micro*, 43(3):18–30, 2023.

Sihao Lin, Zerui Li, Xunyi Zhao, Gengze Zhou, Liuyi Wang, Rong Wei, Rui Tang, Juncheng Li, Hanqing Wang, Jiangmiao Pang, Anton van den Hengel, Jiajun Liu, and Qi Wu. VLNVerse: A benchmark for vision-language navigation with versatile, embodied, realistic simulation and evaluation, 2025. URL https://arxiv.org/abs/2512.19021.

Chang Liu, Henghui Ding, Lingyi Hong, Ning Xu, Linjie Yang, Yuchen Fan, Canyang Wu, Jinrong Zhang, Xusheng He, Ce Bian, Xianjing Han, Jianlong Wu, Mingqi Gao, Sijie Li, Jungong Han, JeongRae Kim, Chaehyun Kim, Changwon Lim, Jungyoon Lee, Gyuil Lim, Doeon Kim, Seong-heum Kim, Pranjal Aggarwal, Sean Welleck, Yiwen Ren, Jianing Liu, Yingxin Wang, Kexin Zhang, Licheng Jiao, Lingling Li, Xu Liu, Jinxing Zhou, Suiyi Zhao, Yanghao Zhou, Ruohao Guo, Liangtao Shi, Jinxia Xie, Xiantao Hu, and Ting Liu. Report of the 8th LSVOS challenge: Complex and multimodal video object segmentation, 2026. URL https://arxiv.org/abs/2609.06078.

Thang Luong and Edward Lockhart. Advanced version of Gemini with Deep Think officially achieves gold-medal standard at the International Mathematical Olympiad, 2025. URL https://deepmind.google/discover/blog/advanced-version-of-gemini-with-deep-think-officially-achieves-gold-medal-standard-at-the-int

NeoCognition. ApprenticeBench, 2026. URL https://neocognition.io/blog/apprentice-bench.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme,

Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024.

OpenAI. Computer use, 2026. URL https://learn.chatgpt.com/docs/computer-use.

OpenAI. Introducing GPT-5.4, 2026. URL https://openai.com/index/introducing-gpt-5-4/.

OpenAI. GPT-6 Astra: A new generation of intelligence, 2026. URL https://openai.com/index/gpt-6-astra/.

OpenAI, Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaiev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, Jerry Tworek, Lorenz Kuhn, Lukasz Kaiser, Mark Chen, Max Schwarzer, Mostafa Rohaninejad, Nat McAleese, o3 contributors, Oleg Mürk, Rhythm Garg, Rui Shu, Szymon Sidor, Vineet Kosaraju, and Wenda Zhou. Competitive programming with large reasoning models, 2025. URL https://arxiv.org/abs/2502.06807.

OpenSUN3D. 7th Workshop on Open-World 3D Scene Understanding and Representations, in conjunction with ECCV 2026, 2026. URL https://opensun3d.github.io/.

Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. Training software engineering agents and verifiers with SWE-Gym. In *International Conference on Machine Learning (ICML)*, 2025.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

Nicholas Pfaff, Thomas Cohn, Sergey Zakharov, Rick Cory, and Russ Tedrake. SceneSmith: Agentic generation of simulation-ready indoor scenes. In *International Conference on Machine Learning (ICML)*, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, Xiaojun Xiao, Kai Cai, Chuang Li, Yaowei Zheng, Chaolin Jin, Chen Li, Xiao Zhou, Minchao Wang, Haoli Chen, Zhaojian Li, Haihua Yang, Haifeng Liu, Feng Lin, Tao Peng, Xin Liu, and Guang Shi. UI-TARS: Pioneering automated GUI interaction with native agents, 2025. URL https://arxiv.org/abs/2501.12326.

Mateo Quiros. Introducing computer use in Gemini 3.5 Flash, 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/introducing-computer-use-gemini-3-5-flash/.

Arijit Ray, Jiafei Duan, Ellis Brown, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A. Plummer, Ranjay Krishna, Kuo-Hao Zeng, and Kate Saenko. SAT: Dynamic spatial aptitude training for multimodal language models. In *Conference on Language Modeling (COLM)*, 2025.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi Robot: Open-ended instruction following with hierarchical vision-language-action models. In *International Conference on Machine Learning (ICML)*, 2025.

Yueqi Song, Frank F. Xu, Shuyan Zhou, and Graham Neubig. Beyond browsing: API-based web agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11066–11085, 2025.

Richard Sutton. The bitter lesson, 2019. URL http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

Feilong Tang, Xiang An, Yunyao Yan, Yin Xie, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Chunyuan Li, Shikun Feng, Changrui Chen, Huajie Tan, Ming Hu, Manyuan Zhang, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. OneVision-Encoder: Codec-aligned sparsity as a foundational principle for multimodal intelligence, 2026. URL https://arxiv.org/abs/2602.08683.

Jingyu Tang, Xu Yang, Zhehao Zhang, Zheng Wang, Qi Cheng, Haozhe Zhao, Manling Li, Wenxuan Li, Xide Xia, and Jiaxin Pei. Realtime-Bench: Can frontier models solve real-time interactive problems?, 2026. URL https://arxiv.org/abs/2610.22155.

Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. RoboGen: Towards unleashing infinite data for automated robot learning via generative simulation. In *International Conference on Machine Learning (ICML)*, 2024.

Bowen Wang, Dunjie Lu, Junli Wang, Tianyi Bai, Shixuan Liu, Zhipeng Zhang, Haiquan Wang, Hao Hu, Tianbao Xie, Shuai Bai, Dayiheng Liu, Que Shen, Junyang Lin, and Tao Yu. CUA-Gym: Scaling verifiable training environments and tasks for computer-use agents, 2026. URL https://arxiv.org/abs/2605.25624.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Tianbao Xie, Mengqi Yuan, Danyang Zhang, Xinzhuang Xiong, Zhennan Shen, Zilong Zhou, Xinyuan Wang, Yanxu Chen, Jiaqi Deng, Junda Chen, Bowen Wang, Haoyuan Wu, Jixuan Chen, Junli Wang, Dunjie Lu, Hao Hu, and Tao Yu. Introducing OSWorld-Verified, 2025. URL https://xlang.ai/blog/osworld-verified.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-Mark prompting unleashes extraordinary visual grounding in GPT-4V, 2023. URL https://arxiv.org/abs/2310.11441.

Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, Yuquan Deng, and Jianfeng Gao. Magma: A foundation model for multimodal AI agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14203–14214, 2025.

John Yang, Kilian Lieret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. SWE-smith: Scaling data for software engineering agents, 2025. URL https://arxiv.org/abs/2504.21798.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. RoboPoint: A vision-language model for spatial affordance prediction for robotics, 2024. URL https://arxiv.org/abs/2406.10721.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. AppAgent: Multimodal agents as smartphone users. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI)*, 2025.

Bojun Zhang, Hangjian Ye, Hao Zheng, Jianzheng Huang, Zhengyu Lin, Zhenhong Guo, and Feng Zheng. PinPoint3D: Fine-grained 3D part segmentation from a few clicks, 2025. URL https://arxiv.org/abs/2509.25970.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. GPT-4V(ision) is a generalist web agent, if grounded. In *International Conference on Machine Learning (ICML)*, 2024.