Scaling Instructable Agents Across Many Simulated Worlds

SIMA Team:[1] **Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, Bethanie Brownfield, Gavin Buttimore, Max Cant, Sarah Chakera, Stephanie C. Y. Chan, Jeff Clune**[1,3]**, Adrian Collister, Vikki Copeman**[2]**, Alex Cullum, Ishita Dasgupta, Dario de Cesare, Julia Di Trapani, Yani Donchev, Emma Dunleavy, Martin Engelcke, Ryan Faulkner, Frankie Garcia, Charles Gbadamosi, Zhitao Gong, Lucy Gonzales**[2]**, Karol Gregor, Arne Olav Hallingstad, Tim Harley, Sam Haves, Felix Hill, Ed Hirst, Drew A. Hudson, Steph Hughes-Fitt, Danilo J. Rezende, Mimi Jasarevic, Laura Kampis, Rosemary Ke, Thomas Keck, Junkyung Kim, Oscar Knagg, Kavya Kopparapu, Andrew Lampinen, Shane Legg, Alexander Lerchner, Marjorie Limont, Yulan Liu, Maria Loks-Thompson, Joseph Marino, Kathryn Martin Cussons**[2]**, Loic Matthey, Siobhan Mcloughlin, Piermaria Mendolicchio, Hamza Merzic, Anna Mitenkova, Alexandre Moufarek, Valeria Oliveira, Yanko Oliveira, Hannah Openshaw, Renke Pan, Aneesh Pappu, Alex Platonov, Ollie Purkiss, David Reichert, John Reid, Pierre Harvey Richemond, Tyson Roberts, Giles Ruscoe, Jaume Sanchez Elias, Tasha Sandars**[2]**, Daniel P. Sawyer, Tim Scholtes, Guy Simmons, Daniel Slater, Hubert Soyer, Heiko Strathmann, Peter Stys, Allison C. Tam**[2]**, Denis Teplyashin, Tayfun Terzi, Davide Vercelli, Bojan Vujatovic, Marcus Wainwright, Jane X. Wang, Zhengdong Wang, Daan Wierstra**[2]**, Duncan Williams, Nathaniel Wong, Sarah York, Nick Young**

[1]Google DeepMind unless otherwise noted, authors listed in alphabetical order, contributions listed at end of report, [2]work performed while at Google DeepMind, [3]University of British Columbia

Building embodied AI systems that can follow arbitrary language instructions in any 3D environment is a key challenge for creating general AI. Accomplishing this goal requires learning to ground language in perception and embodied actions, in order to accomplish complex tasks. The Scalable, Instructable, Multiworld Agent (SIMA) project tackles this by training agents to follow free-form instructions across a diverse range of virtual 3D environments, including curated research environments as well as open-ended, commercial video games. Our goal is to develop an instructable agent that can accomplish anything a human can do in any simulated 3D environment. Our approach focuses on language-driven generality while imposing minimal assumptions. Our agents interact with environments in real-time using a generic, human-like interface: the inputs are image observations and language instructions and the outputs are keyboard-and-mouse actions. This general approach is challenging, but it allows agents to ground language across many visually complex and semantically rich environments while also allowing us to readily run agents in new environments. In this paper we describe our motivation and goal, the initial progress we have made, and promising preliminary results on several diverse research environments and a variety of commercial video games.

Keywords: Agents, Embodiment, Foundation Models, Language, Video Games, 3D Environments

1. Introduction

Despite the impressive capabilities of large language models (Brown et al., 2020; Hoffmann et al., 2022; OpenAI, 2023; Anil et al., 2023; Gemini Team et al., 2023), connecting them to the embodied world that we inhabit remains challenging. Modern AI can write computer programs (Li et al., 2022) or play chess at super-human level (Silver et al., 2018), but the ability of AI to perceive and act in the world remains far below human level. Competence in language alone is easier for AI than grounded perception and behavior, underscoring the well-known paradox that what is easier for AI is harder for humans, and vice versa (Moravec, 1988).

Figure 1 | **Overview of SIMA.** In SIMA, we collect a large and diverse dataset of gameplay from both curated research environments and commercial video games. This dataset is used to train agents to follow open-ended language instructions via pixel inputs and keyboard-and-mouse action outputs. Agents are then evaluated in terms of their behavior across a broad range of skills.

Yet, language is most useful in the abstractions it conveys about the world. Language abstractions can enable efficient learning and generalization (Hill et al., 2020; Colas et al., 2020; Lampinen et al., 2022; Tam et al., 2022; Hu and Clune, 2023). Once learned, language can unlock planning, reasoning (e.g., Huang et al., 2022; Brohan et al., 2023b; Driess et al., 2023; Kim et al., 2023), and communication (Zeng et al., 2022) about grounded situations and tasks. In turn, grounding language in rich environments can make a system's understanding of the language itself more systematic and generalizable (Hill et al., 2019). Thus, several questions emerge: How can we bridge the divide between the symbols of language and their external referents (cf., Harnad, 1990)? How can we connect the abstractions and generality afforded by language to grounded perception and action, and how can we do so in a safe and scalable way?

Here, we draw inspiration from these questions—and the prior and concurrent research projects that have addressed them (e.g., Hermann et al., 2017; Abramson et al., 2020; Brohan et al., 2023a,b; Driess et al., 2023; Wang et al., 2023b; Tan et al., 2024)—to attempt to connect language to grounded behavior at scale. Bridging this gap is a core challenge for developing general *embodied AI*.

The Scalable, Instructable, Multiworld Agent (SIMA) project aims to build a system that can follow *arbitrary* language instructions to act in *any* virtual 3D environment via keyboard-and-mouse actions—from custom-built research environments to a broad range of commercial video games. There is a long history of research in creating agents that can interact with video games or simulated 3D environments (e.g., Mnih et al., 2015; Berner et al., 2019; Vinyals et al., 2019; Baker et al., 2022) and even follow language instructions in a limited range of environments (e.g., Abramson et al., 2020; Lifshitz et al., 2023). In SIMA, however, we are drawing inspiration from the lesson of large language models that training on a broad distribution of data is the most effective way to make progress in general AI (e.g., Brown et al., 2020; Hoffmann et al., 2022; OpenAI, 2023; Anil et al., 2023; Gemini

Team et al., 2023). Thus, in contrast to prior works (e.g., Abramson et al., 2020; Vinyals et al., 2019; Berner et al., 2019; Lifshitz et al., 2023), we are attempting to tackle this problem across many simulated environments, in the most general and scalable way possible, by making few assumptions beyond interacting with the environments in the same way as humans do.

To this end, have made a number of design decisions that make our approach more general, but also more challenging:

- We incorporate many rich, visually complex, open-ended video games containing hundreds of objects in a scene and a large number of possible interactions.
- These environments are asynchronous (e.g., Berner et al., 2019; Vinyals et al., 2019); unlike many research environments, they do not stop and wait while the agent computes its next action.
- Each instance of a commercial video game needs to run on a GPU; thus, we cannot run hundreds or thousands of actors per game per experiment as often done in RL (cf., Espeholt et al., 2018).
- Agents receive the same screen observations that a human playing the game would without access to internal game state, rewards, or any other privileged information (cf., Berner et al., 2019; Vinyals et al., 2019).
- To interact with the environments, agents use the same keyboard-and-mouse controls that humans do (e.g., Baker et al., 2022; Humphreys et al., 2022; Lifshitz et al., 2023), rather than handcrafted action spaces or high-level APIs.
- We focus on following language instructions (e.g., Abramson et al., 2020) rather than simply playing the games to maximize a win-rate or generating plausible behavior (cf., Berner et al., 2019; Vinyals et al., 2019).
- We train and test our agents using open-ended natural language, rather than simplified grammars or command sets (e.g., Abramson et al., 2020).

These design choices make the learning problem harder, but their generality makes expanding to new environments easier: agents use the same interface across environments without requiring a custom design of control and observation spaces for each new game. Furthermore, since the agent-environment interface is human compatible, it allows agents the potential to achieve anything that a human could, and allows direct imitation learning from human behavior. This general interface from language instructions to embodied behavior can also enable agents to transfer previously learned skills zero-shot to never-before-seen games. Doing research in generic virtual environments allows us to test our agents in a broad and challenging range of situations—where the lessons learned are likely to be more applicable to real-world applications with visually rich perception and control such as robotics—without the risks and costs of real-world testing: if the agent crashes a spaceship in a video game, we can just restart the game.

In the SIMA project thus far, we have created an agent that performs short-horizon tasks based on language instructions produced by a user; though instructions could also be produced by a language model (e.g., Jiang et al., 2019; Driess et al., 2023; Wang et al., 2023b; Hu et al., 2023; Ajay et al., 2023). We have a portfolio of over ten 3D environments, consisting of research environments and commercial video games. For research environments we evaluate agents using the ground truth state, but commercial video games are not designed to report on the completion of arbitrary language tasks. We have therefore developed a variety of methods for evaluation in video games, including using optical character recognition (OCR) to detect onscreen text describing task completion, and using human evaluation of recorded videos of agent behavior. In the rest of this tech report, we describe the high-level approach (illustrated in Figure 1) and our initial progress towards the ultimate goal of SIMA: developing an instructable agent that can accomplish anything a human can do in any simulated 3D environment.

2. Related work

SIMA builds on a long history of using games as a platform for AI research. For example, backgammon provided the initial proving ground for early deep reinforcement learning methods (Tesauro et al., 1995), and later works have achieved superhuman performance even in complex board games like Go (Silver et al., 2016, 2018).

Video games Over the last ten years, video games have provided an increasingly important setting for research focused on embodied agents that perform visuomotor control in rich environments. Researchers have used many video game environments, covering a wide spectrum from Atari (Bellemare et al., 2013) to DoTA (Berner et al., 2019) and StarCraft II (Vinyals et al., 2019). In SIMA, however, we restrict our focus to games that resemble 3D physical embodiment most closely, in particular games where the player interacts with a 3D world from a first or over-the-shoulder pseudo-first-person view. This focus excludes many of the games which have previously been used for research, such as the ones listed above. There has however been notable interest in first-person embodied video games as a platform for AI research (Johnson et al., 2016; Tessler et al., 2017; Guss et al., 2019; Pearce and Zhu, 2022; Hafner et al., 2023; Durante et al., 2024; Tan et al., 2024). These video game AI projects have driven the development of many innovative techniques, e.g., learning from videos by annotating them with estimated player keyboard-and-mouse actions using inverse dynamics models (Pearce and Zhu, 2022; Baker et al., 2022). More recently, games that offer API access to the environment have served as a platform for grounding large language models (Wang et al., 2023a), and some works have even considered grounding a language model in a game through direct perception and action of a lower-level controller (Wang et al., 2023b). Instead of focusing on a single game or environment, however, SIMA considers a range of diverse games to train agents on a larger variety of content.

Research environments Other works have focused on custom, controlled environments designed for research. Many of these environments focus on particular domains of real-world knowledge. For example, AI2-THOR (Kolve et al., 2017), VirtualHome (Puig et al., 2018), ProcTHOR (Deitke et al., 2022), AI Habitat (Savva et al., 2019; Szot et al., 2021; Puig et al., 2023), ALFRED (Shridhar et al., 2020), and Behavior (Srivastava et al., 2021) simulate embodied agents behaving in naturalistic rendered scenes. CARLA (Dosovitskiy et al., 2017) provides a simulator for autonomous driving. MuJoCo (Todorov et al., 2012), PyBullet (Coumans and Bai, 2016–2023), and Isaac Gym (Makoviychuk et al., 2021) provide high quality physics simulators for learning low-level control and are used by benchmarks for robotic manipulation such as Meta-World (Yu et al., 2020) and Ravens (Zeng et al., 2021). Albrecht et al. (2022) propose a unified environment encompassing a variety of skills afforded through ecologically-inspired interactions. The Playhouse (Abramson et al., 2020; DeepMind Interactive Agents Team et al., 2021; Abramson et al., 2022a) and WorldLab (e.g., Gulcehre et al., 2019) environments are built using Unity (see Ward et al., 2020). Open Ended Learning Team et al. (2021) and Adaptive Agent Team et al. (2023) also use Unity to instantiate a broad distribution of procedurally generated tasks with shared underlying principles. For the results in this work, we also use Playhouse, WorldLab, and ProcTHOR. In addition, we introduce a new environment, called the Construction Lab.

Robotics Robotics is a key area for research in embodied intelligence. A variety of robotics projects have used simulations for training, to transfer efficiently to real-world robotic deployments (Höfer et al., 2021), though generally within a single, constrained setting. More recent work has focused on environment-generality, including scaling robotic learning datasets across multiple tasks and embodiments (Brohan et al., 2022, 2023a; Stone et al., 2023; Padalkar et al., 2023)—thereby

creating Vision-Language-Action (VLA) models (Brohan et al., 2023a), similar to the SIMA agent. The latter challenge of generalizing or quickly adapting to new embodiments has some parallels to acting in a new 3D environment or computer game where the mechanics are different. Moreover, a variety of recent works have applied pretrained (vision-)language models as a planner for a lower-level instruction-conditional robotic control policy (Brohan et al., 2023b; Driess et al., 2023; Vemprala et al., 2023; Hu et al., 2023). Our approach shares a similar philosophy to the many works that attempt to ground language via robotics. SIMA, however, avoids the additional challenges of costly hardware requirements, resource-intensive data collection, and the practical limitations on diversity of real-world evaluation settings. Instead, SIMA makes progress towards embodied AI by leveraging many simulated environments and commercial video games to obtain the sufficient breadth and richness that we conjecture to be necessary for effectively scaling embodied agents—with the hope that lessons learned (and possibly even the agents themselves) will be applicable to robotic embodiments in the future.

Learning environment models Some works attempt to leverage learned models of environments to train agents in these learned simulations (e.g., Ha and Schmidhuber, 2018; Hafner et al., 2020, 2023; Yang et al., 2023). These methods, however, tend to be difficult to scale to diverse sets of visually complex environments that need to be self-consistent across long periods of time. Nevertheless, learning imperfect models can still be valuable. In SIMA, we build on video models (Villegas et al., 2022), which we fine-tune on game environments. However, we only use the internal state representations of the video models rather than explicit rollouts—in keeping with other approaches that use generative modeling as an objective function for learning state representations (e.g., Gregor et al., 2019; Zolna et al., 2024).

Grounding language Another stream of work—overlapping with those above—has focused on grounding language in simulated 3D environments, through agents that are trained in controlled settings with semi-natural synthetic language (Hermann et al., 2017; Hill et al., 2019), or by imitating human interactions in a virtual house to learn a broader ability to follow natural language instructions (Abramson et al., 2020; DeepMind Interactive Agents Team et al., 2021; Abramson et al., 2022a,b). Moreover, a range of recent works develop agents that connect language to embodied action, generally as part of a hierarchy controlled by a language model (Jiang et al., 2019; Driess et al., 2023; Wang et al., 2023b; Hu et al., 2023; Ajay et al., 2023). We likewise draw inspiration from the idea that language is an ideal interface for directing an agent, but extend our scope beyond the limited affordances of a single controlled environment. In that sense, SIMA overlaps more with several recent works (Reed et al., 2022; Huang et al., 2023; Durante et al., 2024) that also explore training a single model to perform a broad range of tasks involving actions, vision, and language. However, SIMA is distinct in our focus on simultaneously (1) taking a language-first perspective, with all training experiences being language-driven; (2) adopting a unified, human-like interface across environments with language and vision to keyboard-and-mouse control; and (3) exploring a broad range of visually rich, diverse, and human-compatible environments that afford a wide range of complex skills.

Language supports grounded learning, and grounded learning supports language A key motivation of SIMA is the idea that learning language and learning about environments are mutually reinforcing. A variety of studies have found that even when language is not *necessary* for solving a task, learning language can help agents to learn generalizable representations and abstractions, or to learn more efficiently. Language abstractions can accelerate grounded learning, for example accelerating novelty-based exploration in reinforcement learning by providing better state abstractions (Tam et al., 2022; Mu et al., 2022), or composing known goals into new ones (Colas et al., 2020; Nottingham

et al., 2023). Moreover, learning to predict natural-language explanations (Lampinen et al., 2022), descriptions (Kumar et al., 2022), or plans (Hu and Clune, 2023) can help agents to learn more efficiently, and to generalize better out of distribution. Language may be a powerful tool for shaping agent capabilities (Colas et al., 2022).

Conversely, richly grounded learning can also support language learning. Since human language use is deeply integrated with our understanding of grounded situations (McClelland et al., 2020), understanding the subtleties of human language will likely benefit from this grounding. Beyond this theoretical argument, empirical evidence shows that grounding can support even fundamental kinds of generalization—Hill et al. (2019) show that agents grounded in richer, more-embodied environments exhibit more systematic compositional generalization. These findings motivate the possibility that learning both language and its grounding will not only improve grounded actions, but improve a system's knowledge of language itself.

3. Approach

Many overlapping areas of previous and concurrent work share some of our philosophy, motivations, and approaches. What distinguishes the SIMA project is our focus on language-conditional behavior across a diverse range of visually and mechanically complex simulated environments that afford a rich set of skills. In this section, we provide a high-level overview of our approach: our environments, data, agents, and evaluations.

3.1. Environments

SIMA aims to ground language across many rich 3D environments (Figure 2). Thus, we selected 3D embodied environments that offer a broad range of open-ended interactions—such environments afford the possibility of rich and deep language interactions. We focus on environments that are either in a) first-person or b) third-person with the camera over the player's shoulder. To achieve diversity and depth of experience, we use a variety of commercial video games, as well as several environments created specifically for agent research. Each type of environment offers distinct advantages, ranging from open-ended diverse experiences to targeted assessments of agent skills. We have deliberately sought to build a portfolio of games that covers a wide range of settings—from mundane tasks in semi-realistic environments, to acting as a mischevious goat in a world with exaggerated physics, to exploring mythological worlds or science-fiction universes. Below, we briefly describe the environments we have used in SIMA thus far by category and in alphabetical order.

3.1.1. Commercial video games

Commercial video games offer exciting, open-ended worlds full of visual richness and the potential for complex interactions. In SIMA, we have partnered with games developers whose games we used for training agents, and we are continuing to develop relationships with new developers—for our full list of current partners, please see our Acknowledgements section. We focus on a variety of open-world or sandbox games that contain diverse skills, while avoiding games containing harmful content such as extreme violence or biases. We have also sought a broad diversity of worlds and stories, but with a focus on games that exhibit a depth of interesting mechanics. Accordingly, games from our portfolio offer a wide range of distinct challenges in perception and action, from flying a spaceship to mining minerals or crafting armor, as well as more common core features, such as navigation or gathering resources. Games also often include interactions that extend beyond the skillset of typical embodied research environments, such as menu use and interfaces more similar to those faced in computer

control benchmarks (e.g., Humphreys et al., 2022; Koh et al., 2024). For the results in this report, we focus on single-player interactions within these games.

We run instances of each game in a secure Google Cloud environment, using hardware accelerated rendering to a virtual display. This display is streamed to a browser for human gameplay, or to a remote agent client process during evaluation. To instantiate repeatable evaluation or data collection scenarios within each game, we build datasets of save-game files from expert play, and use scripted processes to automate the process of installing game-files, booting the game, navigating its main menu, and loading a specific save-game.

We now provide a brief description of the games we used.

Goat Simulator 3: A third-person game where the player is a goat in a world with exaggerated physics. The player can complete quests, most of which involve wreaking havoc. The goat is able to lick, headbutt, climb, drive, equip a wide range of visual and functional items, and perform various other actions. Throughout the course of the game, the goat unlocks new abilities, such as the ability to fly.

Hydroneer: A first-person mining and base building sandbox where the player is tasked with digging for gold and other resources to turn a profit and enhance their mining operation. To do this, they must build and upgrade their set-ups and increase the complexity and levels of automation until they have a fully automated mining system. Players can also complete quests from non-player characters to craft bespoke objects and gain extra money. Hydroneer requires careful planning and managing of resources.

No Man's Sky: A first- or third-person survival game where the player seeks to explore a galaxy full of procedurally-generated planets. This involves flying between planets to gather resources, trade, build bases, and craft items that are needed to upgrade their equipment and spaceship while surviving a hazardous environment. No Man's Sky includes a large amount of visual diversity—which poses important challenges for agent perception—and rich interactions and skills.

Satisfactory: A first-person, open-world exploration and factory building game, in which players attempt to build a space elevator on an alien planet. This requires building increasingly complex production chains to extract natural resources and convert them into industrial goods, tools, and structures—whilst navigating increasingly hostile areas of a large open environment.

Teardown: A first-person, sandbox–puzzle game in a fully destructible voxel world where players are tasked with completing heists to gain money, acquiring better tools, and undertaking even more high-risk heists. Each heist is a unique scenario in one of a variety of locations where players must assess the situation, plan the execution of their mission, avoid triggering alarms, and escape before a timer expires. Teardown involves planning and using the environment to one's advantage to complete the tasks with precision and speed.

Valheim: A third-person survival and sandbox game in a world inspired by Norse mythology. Players must explore various biomes, gather resources, hunt animals, build shelter, craft equipment, sail the oceans and defeat mythological monsters to advance in the game—while surviving challenges like hunger and cold.

Wobbly Life: A third-person, open-world sandbox game where the player can explore the world, unlock secrets, and complete various jobs to earn money and buy items, leading up to buying their own house. They must complete these jobs whilst contending with the rag-doll physics of their characters and competing against the clock. The jobs require timing, planning, and precision to be completed. The world is extensive and varied, with a diverse range of interactive objects.

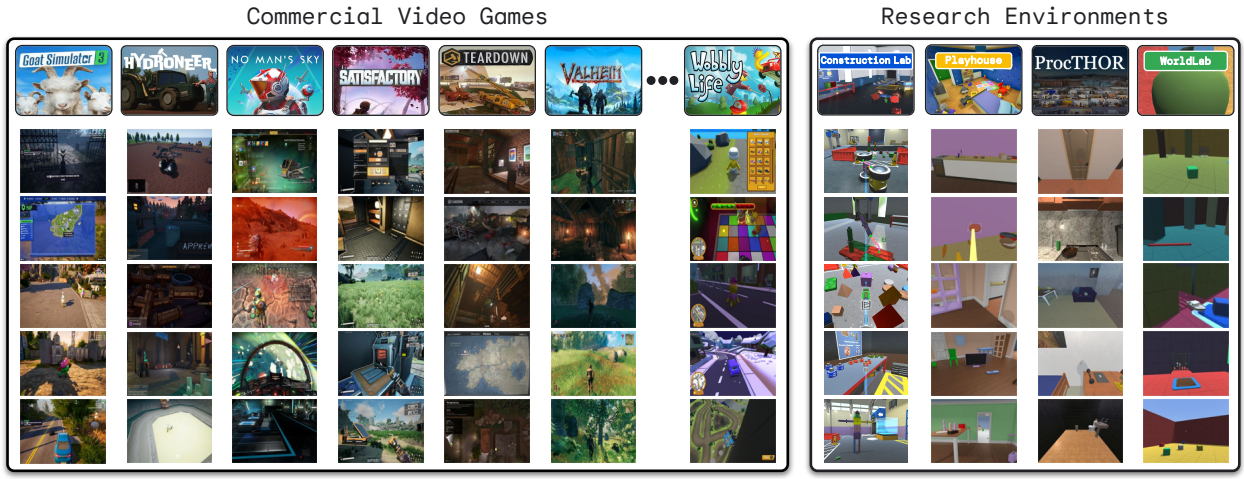

Figure 2 | **Environments.** We use over ten 3D environments in SIMA, consisting of commercial video games and research environments. The diversity of these environments is seen in their wide range of visual observations and environmental affordances. Yet, because these are all 3D environments, basic aspects of 3D embodied interaction, such as navigation, are shared. Commercial video games offer a higher degree of rich interactions and visual fidelity, while research environments serve as a useful testbed for probing agent capabilities.

3.1.2. Research environments

In contrast to commercial video games, AI research environments are typically more controllable, offering the ability to instill and carefully assess particular skills, and more rapid and reliable evaluations of task completion. Unlike many of the games in our portfolio, several of these research environments also tend to feature more real-world analogous—if still simplified—physical interactions.

We have drawn on several prior research environments and developed a new environment—the Construction Lab—that incorporates important challenges which were not otherwise well-captured by our other environments.

Construction Lab: A new research environment where agents need to build novel items and sculptures from interconnecting building blocks, including ramps to climb, bridges to cross, and dynamic contraptions. Construction Lab focuses on cognitive capabilities such as object manipulation and an intuitive understanding of the physical world.

Playhouse: An environment used in various prior works (Abramson et al., 2020; DeepMind Interactive Agents Team et al., 2021; Abramson et al., 2022a), consisting of a procedurally-generated house environment with various objects. We have augmented this environment with improved graphics and richer interactions, including skills like cooking or painting.

ProcTHOR: An environment consisting of procedurally-generated rooms with realistic contents, such as offices and libraries, introduced by Deitke et al. (2022). Although benchmark task sets exist in this environment, prior works have not used keyboard and mouse actions for agents; thus we focus on this environment primarily for data collection rather than evaluation.

WorldLab: An environment used in prior work (e.g., Gulcehre et al., 2019), further specialized for testing embodied agents by using a limited set of intuitive mechanics, such as sensors and doors, and relying primarily on the use of simulated physics on a range of objects.

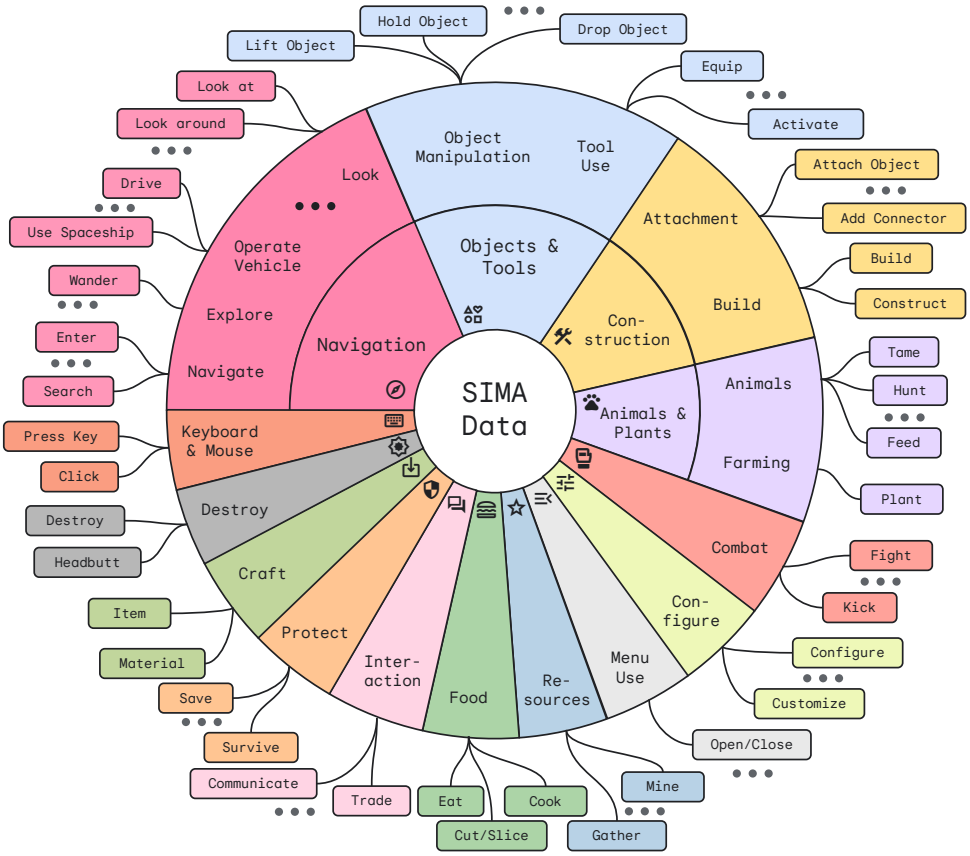

Figure 3 | **Instructions Across SIMA Data.** The SIMA dataset includes a broad range of text instructions that can be roughly clustered into a hierarchy. Due to the common 3D embodied nature of the environments that we consider, many generic tasks, such as navigation and object manipulation, are present in multiple environments. Categories were derived from a data-driven hierarchical clustering analysis of the human-generated text instructions within a fixed, pretrained word embedding space. Note that the area of each cluster in the wheel in Figure 3 does not correspond to the exact number of instructions from that cluster in the dataset.

3.2. Data

Our approach relies on training agents at scale via behavioral cloning, i.e., supervised learning of the mapping from observations to actions on data generated by humans. Thus, a major focus of our effort is on collecting and incorporating gameplay data from human experts. This includes videos, language instructions and dialogue, recorded actions, and various annotations such as descriptions or marks of success or failure. These data constitute a rich, multi-modal dataset of embodied interaction within over 10 simulated environments, with more to come.[1] Our data can be used to augment and leverage existing training data (e.g., Abramson et al., 2020), or to fine-tune pretrained models to endow them with more situated understanding. These datasets cover a broad range of instructed tasks: Figure 3 shows instruction clusters derived from hierarchically clustering the text instructions present in the data within a fixed, pretrained word embedding space.

Yet, collecting data at scale is not sufficient for training successful agents. Data quality processes

[1]Note: Due to a limited amount of collected data and/or evaluations, we present agent evaluation results (Section 4) on a subset of 7 of these environments.

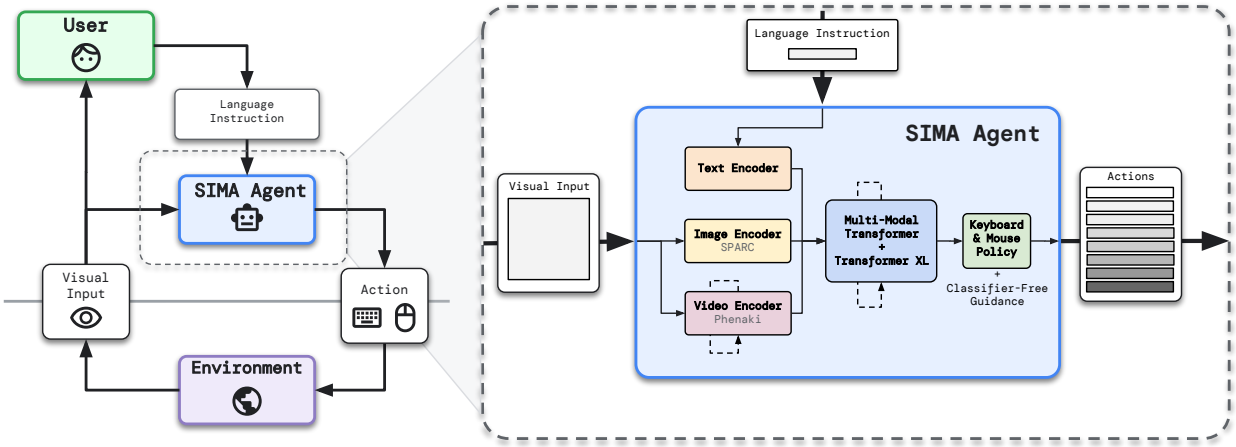

Figure 4 | **Setup & SIMA Agent Architecture.** The SIMA agent receives language instructions from a user and image observations from the environment, and maps them to keyboard-and-mouse actions.

are critical to ensuring an accurate and unconfounded mapping between language and behavior. This presents various technical challenges. We take care to engineer our data collections, including preprocessing and filtering the raw data, to highlight important skills and effectively train our agents.

Data collections We collect data using a variety of methods, including allowing single players to freely play, and then annotating these trajectories with instructions post-hoc. We also perform two-player setter-solver collections (Abramson et al., 2020; DeepMind Interactive Agents Team et al., 2021), in which one player instructs another what to do in selected scenarios while sharing a single player view in order to match the single-player collections. All our data collections were performed with participants contracting with Google. The full details of our data collection protocols, including compensation rates, were reviewed and approved by an independent Human Behavioral Research Committee for ethics and privacy. All participants provided informed consent prior to completing tasks and were reimbursed for their time.

Preprocessing, filtering, and weighting Before training, we perform a variety of offline preprocessing steps, including resizing data for agent input, filtering out low-quality data using a variety of heuristics, and remixing and weighting data across environments and collections to prioritize the most effective learning experiences.

3.3. Agent

The SIMA agent maps visual observations and language instructions to keyboard-and-mouse actions (Figure 4). Given the complexity of this undertaking—such as the high dimensionality of the input and output spaces, and the breadth of possible instructions over long timescales—we predominantly focus on training the agent to perform instructions that can be completed in less than approximately 10 seconds. Breaking tasks into simpler sub-tasks enables their reuse across different settings and entirely different environments, given an appropriate sequence of instructions from the user.

Our agent architecture builds on prior related work (Abramson et al., 2020, 2022a), but with various changes and adaptations to our more general goals. First, our agent incorporates not only trained-from-scratch components, but also several pretrained models—including a model trained

on fine-grained image-text alignment, SPARC (Bica et al., 2024), and a video prediction model, Phenaki (Villegas et al., 2022)—which we further fine-tune on our data through behavioral cloning and video prediction, respectively. In preliminary experiments, we found that these models offer complementary benefits. Combining these pre-trained models with fine-tuning and from-scratch training allows the agent to utilize internet-scale pretraining while still specializing to particular aspects of the environments and the control tasks that it encounters.

More specifically, our agent (Figure 4) utilizes trained-from-scratch transformers that cross-attend to the different pretrained vision components, the encoded language instruction, and a Transformer-XL (Dai et al., 2019) that attends to past memory states to construct a state representation. The resulting state representation is provided as input to a policy network that produces keyboard-and-mouse actions for sequences of 8 actions. We train this agent with behavioral cloning, as well as an auxiliary objective of predicting goal completion.

We use Classifier-Free Guidance (CFG; Ho and Salimans, 2022; Lifshitz et al., 2023) to improve the language-conditionality of a trained agent when running it in an environment. CFG was originally proposed for strengthening text-conditioning in diffusion models (Ho and Salimans, 2022), but has also proven useful for similar purposes with language models (Sanchez et al., 2023) and language-conditioned agents (Lifshitz et al., 2023). That is, we compute the policy, π, with and without language conditioning, and shift the policy logits in the direction of the difference between the two:

$$\pi_{CFG} = \pi\left(\text{image}, \text{language}\right) + \lambda\left(\pi\left(\text{image}, \text{language}\right) - \pi\left(\text{image}, \cdot\right)\right).$$

3.4. Evaluation methods

Our focus on generality in SIMA introduces challenges for evaluation. While research environments may provide automated methods for assessing whether language-following tasks have been successfully completed, such success criteria may not be generally available. That is, language instructions may not correspond to goal states recorded by an environment (e.g. a user might instruct *"make a pile of rocks to mark this spot"* or *"see if you can jump over this chasm"*).

Evaluating agents in commercial video games poses substantial additional challenges. Video game evaluations cannot rely on access to privileged information about the state of an environment. Additionally, it is difficult to reinstate agents in precisely the same state in environments that are not designed as reproducible benchmarks, and loading each task in commercial video games is considerably slower and more costly than those in research environments. Achieving fast, stable, and reliable evaluations comparable across environments is thus challenging. We therefore use a range of distinct evaluation types that provide different trade-offs in efficiency, cost, accuracy, and coverage.

Moreover, ensuring that our evaluations truly assess language conditionality, rather than environmental affordances, requires care. For instance, if a task contains a knife, a cutting board, and a carrot, the agent may ascertain the goal (*"cut the carrot on the cutting board"*) without relying on the language instruction. Thus, task settings need to afford a diversity of actions, ideally testing multiple instructions from a single initial state, to properly evaluate whether the agent's actions are driven by language.

Action log-probabilities One simple approach is to evaluate agents based on their action predictions on held-out evaluation data. However, consistent with prior findings (Abramson et al., 2022b; Baker et al., 2022), we observed that agent action log-probabilities on evaluation data show at most a weak correlation with agent performance beyond the most basic skills. Thus, online evaluations, in which

the agent interacts with the environment, are needed to understand agent performance in detail.

Static visual input Similar to predicting actions on held-out data, we can provide the agent with a static visual input and a language instruction to perform a particular valid action (e.g., *"jump"*) to assess simple responses directly mapping to particular keyboard and/or mouse actions. We have used evaluations of this form for our commercial video game environments, as they have the advantage of not requiring actually loading a game. While these evaluations can be a useful early signal, they do not reliably predict success on prolonged tasks.

Ground-truth Our internally-developed research environments (Construction Lab, Playhouse, and WorldLab) are capable of providing ground-truth assessments of whether language-following tasks have been successfully completed. These tasks can depend on the state of the agent (*"move forward"*) and the surrounding environment (*"lift the green cube"*), as well as more complex interactions (*"attach a connector point to the top of the large block"* or *"use the knife to chop the carrots"*). Such tasks enable robust testing of a range of particular skills, with a highly reliable signal of task success. Moreover, we design the task settings and evaluation to be strong tests of precision; for example, many tasks include distractor objects, for which the episode is marked as an immediate failure if the agent interacts with the distractors rather than the instruction target—even if the agent might have completed the actual task later. We also include other types of assessments, such as instructing the agent to complete one goal, and then interrupting with another goal to evaluate whether it switches appropriately—this ensures that agents are sufficiently responsive to changes in commands. A subset of our research environment tasks are used to provide a fast evaluation signal of agent progress during training.

Optical character recognition (OCR) Many of our commercial video game environments provide on-screen text signalling the completion of tasks or quests, or even the results of lower-level actions like collecting resources or entering certain areas of a game. By detecting on-screen text using OCR in pre-defined evaluation scenarios, sometimes in combination with detecting specific keyboard-and-mouse actions, we can cheaply assess whether the agent has successfully performed particular tasks. This form of automated evaluation also avoids the subjectivity of human evaluations. We make use of OCR evaluation in particular for two games, No Man's Sky and Valheim, which both feature a significant amount of on-screen text. In No Man's Sky, for example, we have developed evaluation tasks such as *"mine carbon/salt/ferrite"*, *"use the analysis visor"*, or *"open the exosuit menu"*. Similarly, in Valheim we have tasks such as *"collect wood/stone/raspberries"*, *"use the workbench"*, or *"cook food"*. In general, however, OCR evaluations are restricted to tasks that signal completion with game-specific text rather than arbitrary tasks that can be specified with language instructions and which we would expect a general agent to be able to solve. Other video games also have significantly less on-screen text, which makes the range of behaviors that can be evaluated in these games with OCR very narrow.

Human evaluation In the many cases where we cannot automatically derive a signal of task success, we turn to humans to provide this assessment. While this is our most general evaluation method, it is also the slowest and most expensive. We use human judges who are game experts, i.e., they have played these specific games for at least 16 hours, and often over the course of several weeks. We ask them to review recorded agent videos, collecting multiple ratings of the same video from different judges (typically 5) to ensure reliable assessments. We also encourage strict evaluations: we instruct judges to mark an episode as a failure in cases where the agent performs irrelevant actions first, even if the agent successfully completes the instructed task afterward.

We curated our human-evaluation tasks by identifying a list of frequently-occurring verbs in English, and combined it with a list of verbs that naturally emerged from gameplay and interactive testing of our agents. We use this verb list as a foundation for our evaluations across all video game environments. We assign each task (save state and instruction pair) to a single, most-representative skill category (e.g. "craft items"), even though most tasks require a wide range of implicit skills to succeed (e.g. crafting often requires menu use). The resulting evaluation set provides a long term challenge for agent research that spans a wide range of difficulties—from simple game agnostic tasks such as *"turn left"*, to ones testing specialized game knowledge *"compare the crafting cost of antimatter and antimatter housing"*, to ones utilising broader semantic knowledge such as *"take the pitchfork from the person shoveling hay"*. Grounding our evaluation framework in the distribution of natural language allows us to test our agents in both common and adversarial scenarios, and thereby to measure our progress towards our long-term goal of developing an instructable agent that can accomplish anything a human can do in any simulated 3D environment.

In the results below (Section 4), we primarily report evaluation scores based on ground-truth evaluations for research environments and combined OCR and human evaluations for commercial video game environments. Across the 7 environments for which we have evaluations, we have a total of 1,485 unique tasks, spanning a range of 9 skill categories, from movement (*"go ahead", "look up", "jump"*) to navigation (*"go to the HUB terminal", "go to your ship"*), resource gathering (*"collect carbon", "get raspberries"*), object management (*"use the analysis visor", "cut the potato"*), and more. (For reference, MineDojo (Fan et al., 2022), a related work investigating language-conditional agents in MineCraft, used 1,581 unique tasks spanning 4 skill categories: survival, harvest, tech-free, and combat). Given the diversity and coverage of our current evaluations, they provide a reasonable assessment of the fundamental language-conditional skills that we expect from our agent. Yet, there remains ongoing work in developing more scalable, general, and reliable evaluations, particularly as we move toward more complex and open-ended tasks.

3.4.1. Latency mitigations

Our agent is evaluated in several environments that run in real-time, asynchronously to the agent. This can pose challenges for the timely execution of agent-generated actions. Latencies or delays (Bratko et al., 1995) are introduced by the computation of actions and the transmission of observations and actions over the network. We account for this latency during behavioral cloning by predicting actions that are offset in time relative to the visual input to the agent, and mirror this offset during evaluation by appropriate buffering of observations and actions during neural-network inference. We additionally minimize latencies with appropriate scheduling of action computation on TPU accelerators, on-device caching of neural-network state across timesteps, and by careful choices of batch size and other implementation details.

3.5. Responsibility

We follow a structured approach to responsible model development, to identify, measure, and manage foreseeable ethics and safety challenges. These are informed by academic literature reviews, engaging with internal ethics teams, and developing comprehensive ethical assessments that document key risks with mitigation strategies. We ensure that our research projects uphold Google's AI Principles.[2] SIMA was carefully assessed and reviewed to ensure that its societal benefits outweigh the risks, and that appropriate risk mitigations are incorporated.

[2] https://ai.google/responsibility/principles/

Figure 5 | **Agent Trajectories.** The SIMA agent is capable of performing a range of language-instructed tasks across diverse 3D virtual environments. Here, we provide several representative, visually salient examples of the agent's capabilities that demonstrate basic navigation and tool use skills.

Benefits SIMA is a cutting-edge research initiative which focuses on how to develop instructable agents in simulated environments. This research presents interesting opportunities for the future of humans and AI collaborating together; unlike LLMs, SIMA is able to both understand natural language instructions and dynamic, interactive 3D environments. This presents a new paradigm for working with AI agents, and the potential for exciting new immersive 3D experiences with AI. Finally, simulated environments present a safer alternative for research compared to other AI deployments.

Risks As well as these benefits, we have reflected on potential risks associated with training on video game data. These include risks associated with training an agent on games that include violent, explicit or otherwise harmful behaviors. We have also reflected on the implications on representational harms, as the agent may learn from stereotyped depictions or actions in game settings. Besides these risks, there are also down stream risks associated with the future hypothetical deployments of SIMA, through either intentional misuse or benign action.

Mitigations We have worked to ameliorate these risks through a holistic approach, including:

- Careful curation of content. We avoided a number of games that have scientifically interesting, but violent environments. We also outlined behavioral "red-lines" with our ethics and safety teams; games with content that violates these red-lines are not used.
- Continuous evaluations of SIMA's safety performance.

- Ensuring SIMA's deployments and agreements are transparent, and for now remain in a controlled, closed environment.

Ultimately, given the careful training data selection and constrained deployment environment of SIMA, we are confident we can maximize the benefits while minimising the ethical risks.

4. Initial results

In this section, we report initial evaluation results of the SIMA agent. After presenting several qualitative examples of the SIMA agent's capabilities, we start by considering the quantitative performance of the SIMA agent, broken down by environment and skill category. We then compare these results with several baselines and ablations, allowing us to assess the generalization capabilities of the agent and the efficacy of our design choices. Finally, we investigate a subset of evaluation tasks to estimate human-level performance as an additional comparison.

Qualitative examples To provide a sense of the agent's general capabilities, Figure 5 displays several representative examples of the agent in our commercial video game environments. Despite the visual diversity of the environments, the agent is capable of performing these tasks, demonstrating basic navigation and tool use skills. Even when the instructed target is not in view (*"go to the spaceship"* and *"go to the HUB"*), the agent is able to find the target. For further qualitative examples, please refer to the accompanying website.[3]

4.1. Performance across environments and skills

In Figure 6, we report the average performance of the SIMA agent across the seven environments for which we have quantitative evaluations. Averages are calculated across multiple episodes per task (in research environments, one episode per task in video games), multiple tasks per environment, and across three training runs with different random seeds. Error bars denote the 95% confidence intervals (CIs) across the tasks within that environment and the three training runs with different random seeds. We note that developing informative evaluation tasks is in itself an ongoing effort, and the quantitative results in this work reflect only the range of particular behaviors that are evaluated at this point in time.

Overall, the results show that the SIMA agent is able to complete a range of tasks across many environments, but there remains substantial room for improvement. Performance is better for Playhouse and WorldLab, which are comparatively simpler research environments. For the more complex commercial video game environments, we see that performance is, understandably, somewhat lower. Notably, performance on Construction Lab is lower as well, highlighting the relative difficulty of this research environment and its evaluation tasks. This enables the SIMA platform to serve as a useful testbed for further development of agents that can connect language to perception and action.

In order to better understand the performance of the SIMA agent across an increasing variety of simulated environments, we developed an evaluation framework grounded in natural language for adding and clustering evaluation tasks, as detailed in our evaluation methods. As these skill clusters are derived from our evaluation tasks rather than the training data, they are similar to, yet distinct from, those in Figure 3. As shown in Figure 7, performance varies across different skill categories, including within skill clusters such as "movement" or "game progression". Note that even seemingly simple skill clusters can involve nontrivial game interactions, e.g., some of the "look" tasks involve

[3] https://deepmind.google/discover/blog/sima-generalist-ai-agent-for-3d-virtual-environments/

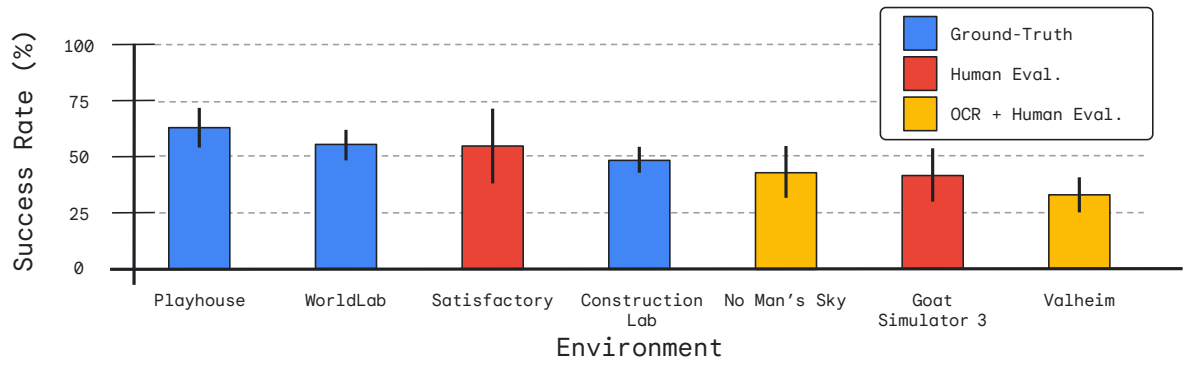

Figure 6 | **Average Success Rate of the SIMA Agent by Environment.** Agents achieve notable success, but are far from perfect; their success rates vary by environment. Colors indicate the evaluation method(s) used to assess performance for that environment. (Note that humans would also find some of these tasks challenging, and thus human-level performance would not be 100%, see Section 4.3.)

skills like steering a spaceship (*"look at a planet"*) or orienting based on the surrounding terrain (*"look downhill"*). While there are many subtleties depending on these additional interactions and the mechanics of the environment in which the skill is used, in general, skills that require more precise actions or spatial understanding ("combat", "use tools", "build") tend to be more challenging.

4.2. Evaluating environment generalization & ablations

We compare our main SIMA agent to various baselines and ablations, both in aggregate (Figure 8) and broken down across our environments (Figure 9). The agents we report across all environments include:

- **SIMA:** Our main SIMA agent, which is trained across all environments except for Hydroneer and Wobbly Life, which we use for qualitative zero-shot evaluation.
- **Zero-shot:** Separate SIMA agents trained like the main agent, but only on $N - 1$ of our environments, and evaluated zero-shot on the held-out environment—that is, without any BC training on it. These agents assess the transfer ability of our agent in a controlled setting. (Note that these agents use the same pretrained encoders as the main SIMA agent, which were finetuned on data from a subset of our environments; thus, in some cases the pretrained encoders will have been tuned with visual inputs from the held-out environment, even though the agent has not been trained to act in that environment. However, the encoders were not fine-tuned on data from Goat Simulator 3, thus the transfer results in that case are unconfounded.)
- **No pretraining ablation:** An agent where we removed the pretrained encoders in the SIMA agent. We replaced these models with a ResNet vision model that is trained from scratch (as in Abramson et al., 2022a), as in preliminary experiments we found training the SPARC/Phenaki encoders through agent training resulted in poor performance. Comparing to this agent tests the benefits of pretrained models for agent performance.
- **No language ablation:** An agent that lacks language inputs, during training as well as evaluation. Comparing to this agent shows the degree to which our agent's performance can be explained by simple language-agnostic behavioral priors.
- **Environment-specialized:** We additionally train an expert agent on each environment, which is trained only on data corresponding to that environment, but still includes the more broadly pretrained encoders. We normalize the performance of all other agents by the expert agent on

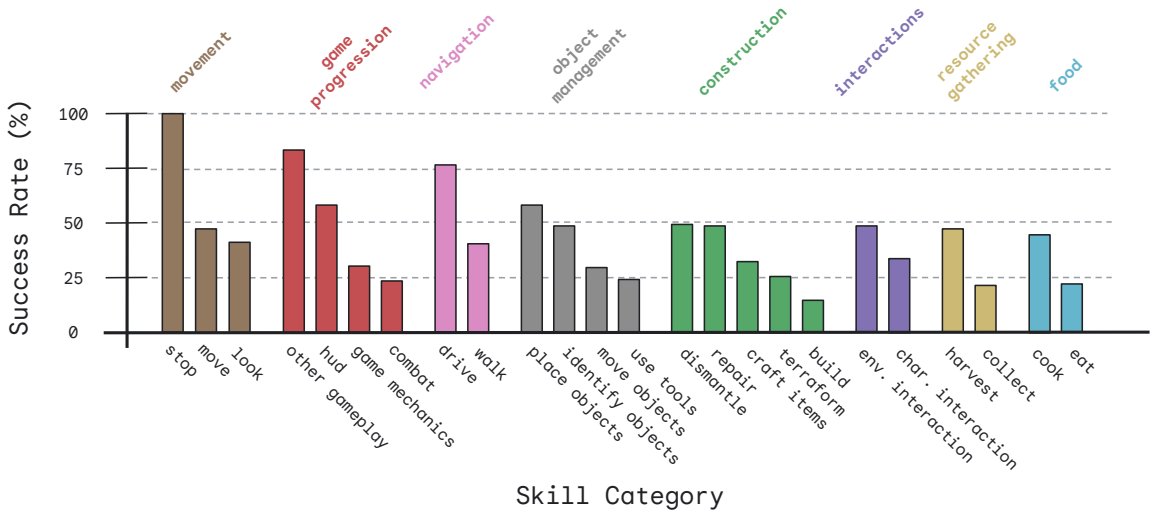

Figure 7 | **Average Success Rate of the SIMA Agent by Skill Category**. Agents exhibit varying degrees of performance across the diverse skills that we evaluate, performing some skills reliably and others with more limited success. Skill categories are grouped into clusters (color), which are derived from our evaluation tasks.

each environment, as a measure of what is possible using our methods and the data we have for that environment.

Note that due to the number of comparison agents, we only ran a single seed for each, rather than the three seeds used for the main SIMA agent. Each agent is evaluated after 1.2 million training steps.[4] The bars in Figure 8 and Figure 9 represent average performance (normalized relative to the environment-specialist); the errorbars are parametric 95%-CIs across tasks and seeds (where multiple seeds are available).

Figure 8 shows a summary of our results, while Figure 9 shows the results by environment. SIMA outperforms environment-specialized agents overall (67% average improvement over environment-specialized agent performance), thus demonstrating positive transfer across environments. We statistically quantify this benefit by using a permutation test on the mean difference across the per-task performance of the SIMA agent and the environment-specialized agent within each domain; in every case SIMA significantly outperforms the environment-specialized agent (p-values on each environment respectively: 0.001, 0.002, 0.036, 0.0002, 0.008, 0.004, and 0.0002). Furthermore, SIMA performs much better than the baselines. SIMA substantially outperforms the no-pretraining baseline overall (permutation test $p < 0.001$), thus showing that internet-scale knowledge supports grounded learning—though the magnitude and significance of the benefit varies across the environments (permutation test p-values respectively 0.0002, 0.14, 0.041, 0.0002, 0.244, 0.052, 0.032). Finally, the no-language ablation performs very poorly (all permutation tests $p < 0.001$). Importantly, this demonstrates not only that our agent *is in fact* using language, but also that our evaluation tasks are effectively designed to test this capability, rather than being solvable by simply executing plausible behaviors.

[4]With one exception: as we had a relatively small quantity of data for Goat Simulator 3, we attempted to prevent the environment-specialized baseline from overfitting by evaluating it every 200,000 training steps, then selecting the best performing number of steps, which was 400,000 steps, as our environment-specialized baseline. Although this is a biased selection process, because we are using the environment-specialized agent as a baseline, it will only lead to *underestimating* the advantage of SIMA.

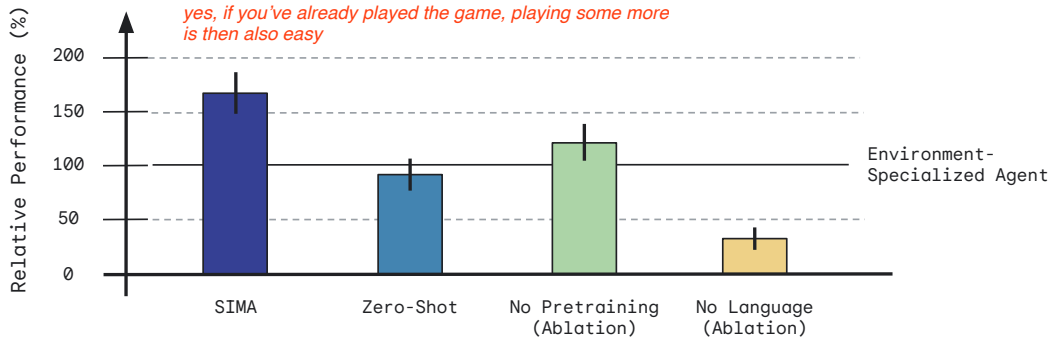

Figure 8 | **Aggregate Relative Performance.** Bars indicate the performance of the SIMA agent as well as the baselines and ablations relative to the performance of the environment-specialized agents, aggregated equally across environments. The SIMA agent outperforms ablations that do not incorporate internet pretraining and substantially outperforms an ablation without language. The solid line shows environment-specialized relative performance, which by normalization is 100%.

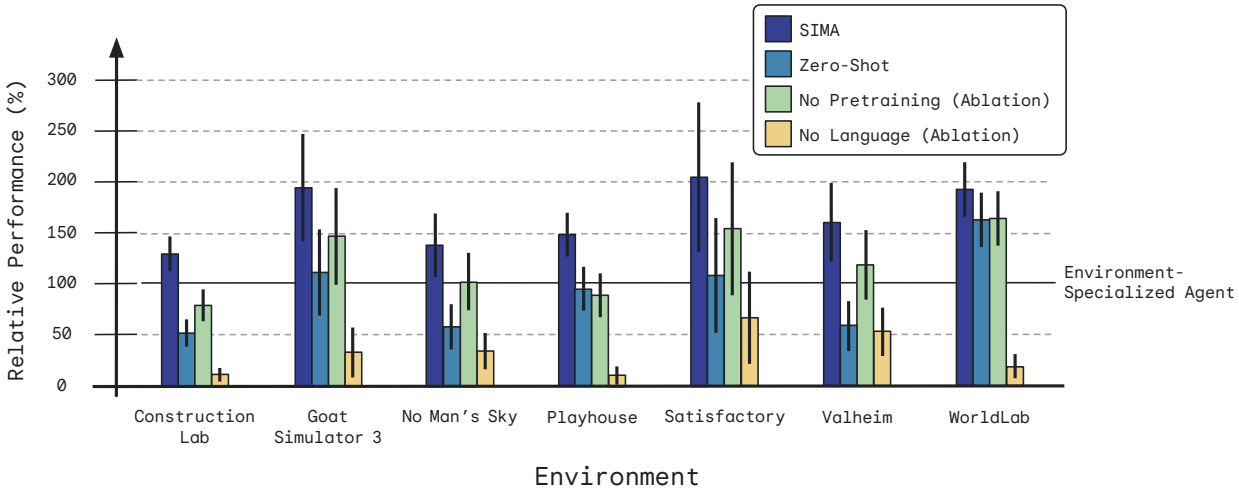

Figure 9 | **Per-Environment Relative Performance.** Bars indicate the performance of the SIMA agent as well as the baselines and ablations relative to the performance of the environment-specialized agents. While performance varies across the environments, the general pattern of results is largely preserved. Even when trained while holding out an environment and evaluated zero-shot on the unseen environment, our agent can achieve non-trivial performance—almost always outperforming the no-language ablation, and in some cases even matching or exceeding environment-specialized agent performance. The solid line shows the relative performance of an environment-specialized agent, which by normalization is 100%.

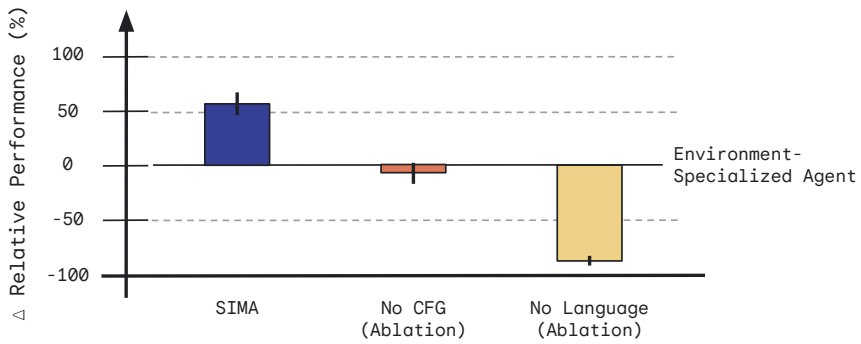

Figure 10 | **Evaluating the Benefit of Classifier-Free Guidance.** Comparing the SIMA agent to an ablation without classifier-free guidance (CFG), CFG substantially improves language conditionality. However, even without CFG, the agent still exhibits language-conditional behavior, outperforming the No Language ablation. Note that this evaluation was performed only on a subset of our research environments: Construction Lab, Playhouse, and WorldLab.

The zero-shot evaluations are also promising. Even when tested in an environment on which it has not been trained to act the agent demonstrates strong performance on general tasks, though of course it falls short in achieving environment-specific skills. Zero-shot agents are capable of performing generic navigation skills that appear across many games (e.g. "go down the hill"), and show some more complex abilities like grabbing an object by its color, using the fact that color is consistent across games, and the consistent pattern that most games use left mouse to grab or interact with objects. Importantly, even on the Goat Simulator 3 environment, where the agents have not even received visual finetuning, the zero-shot agent still performs comparably to the environment-specialized one—thus showing transfer is not driven by the visual components alone. Note that even where the numerical performance of the zero-shot and environment-specialized agents is similar, they are generally good at different skills—with the environment-specialized agent performing well on game-specific interactions, but performing more weakly on common skills that are supported across many games, and that the zero-shot agent therefore can execute.

Note that zero-shot performance is especially strong on the WorldLab environment for three reasons. First, the evaluation tasks for this environment contain a relatively larger proportion of domain-general skills, such as recognizing objects by color, because we use them as rapid tests of agent capabilities. Second, this environment uses the same underlying engine and shares some implementation details with the other internal research environments, which may support behavioral transfer despite their varied visual styles, asset libraries, physical mechanics, and environment affordances. Furthermore, environment-specialized agent performance may be slightly weaker on this environment because there is a non-trivial distribution shift from training to test. This is because some of our data comes from earlier versions of the environment with differences in dynamics, and task distributions. Agents trained across multiple environments may be more robust to this distribution shift.

Classifier-free guidance Finally, Figure 10 compares the performance of agents with and without classifier-free guidance (CFG; Lifshitz et al., 2023), evaluated on a subset of our research environments: Construction Lab, Playhouse, and WorldLab. Without CFG ($\lambda = 0$), the SIMA agent performs noticeably worse. However, the No CFG agent still exhibits a high degree of language conditionality, significantly outperforming the No Language baseline. These results show the benefit of CFG, highlighting the impact that inference-time interventions can have on agent controllability.

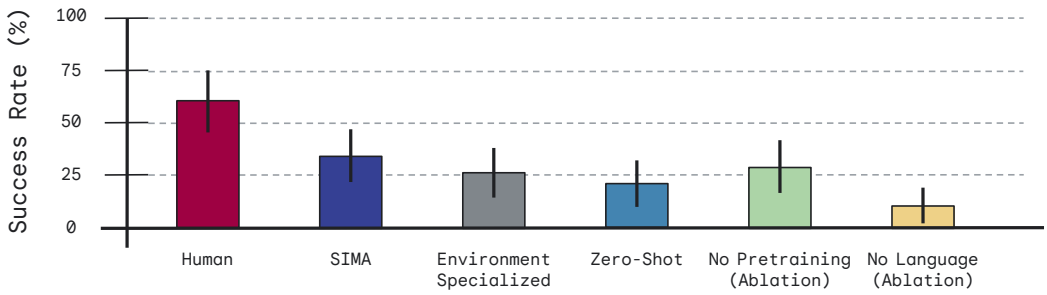

Figure 11 | **Comparison with Human Performance on No Man's Sky.** Evaluating on a subset of tasks from No Man's Sky, human game experts outperform all agents. Yet, humans only achieve 60% success on this evaluation. This highlights the difficulty of the tasks considered in this project.

4.3. Human comparison

To provide an additional baseline comparison, we evaluated our agents against expert human performance on an additional set of tasks from No Man's Sky, which were chosen to test a focused set of skills in a diverse range of settings. These tasks range in difficulty, from simple instructions (*"walk forward"*) to more complex instructions (*"use the analysis visor to identify new animals"*). The humans who performed the tasks were players who participated in our data collection and had experience with the game. We evaluated human performance using the same judges and evaluation setup that was used for our agents; the judges were not told that they were evaluating human performance rather than agents.

Results are summarized in Figure 11 with error bars denoting parametric 95%-CIs. The human players achieved a success rate of only 60% on these tasks, demonstrating the difficulty of the tasks we considered in this project and the stringency of our evaluation criteria. For example, some human failures appear to be due to engaging in unnecessary behaviors before completing the task, like initially opening and interacting with the starship menu when instructed to *"recharge the mining beam,"* or entering analysis mode after scanning when told to *"mine oxygen."* Despite these challenging evaluations, the SIMA agent achieved non-trivial performance (34% success), far exceeding that of the No Language baseline (11% success), for example. We note that 100% success may not necessarily be achievable, due to disagreement between human judges on more ambiguous tasks. Nevertheless, there is still considerable progress needed to match human performance. This underscores the utility of the entire SIMA setup for providing a challenging, yet informative, metric for assessing grounded language interactions in embodied agents.

5. Looking ahead

SIMA is a work in progress. In this tech report, we have described our goal and philosophy, and presented some preliminary results showing our agent's ability to ground language instructions in behavior across a variety of rich 3D environments. We see notable performance and early signs of transfer across environments, as well as zero-shot transfer of basic skills to held-out environments. Still, many skills and tasks remain out of reach. In our future work, we aim to **a)** scale to more environments and datasets by continuing to expand our portfolio of games, environments, and datasets; **b)** increase the robustness and controllability of agents; **c)** leverage increasingly high-quality pretrained models (Gemini Team et al., 2023); and **d)** develop more comprehensive and carefully controlled evaluations.

We believe that by doing so, we will make SIMA an ideal platform for doing cutting-edge research on grounding language and pretrained models safely in complex environments, thereby helping to tackle a fundamental challenge of AGI. Our research also has the potential to enrich the learning experiences and deployment environments of future foundation models; one of our goals is to ground the abstract capabilities of large language models in embodied environments. We hope that SIMA will help us learn how to overcome the fundamental challenge of linking language to perception and action at scale, and we are excited to share more details about our research in the future.

Acknowledgements

We thank the following games developers for partnering with us on this project: Coffee Stain, Foulball Hangover, Hello Games, Keen Software House, Rubberband Games, Saber Interactive / Tuxedo Labs, and Strange Loop Games. We also thank Bica et al. (2024) for their assistance in incorporating SPARC into the SIMA agent as well as Zolna et al. (2024) and Scott Reed for their assistance in incorporating Phenaki into the SIMA agent. We thank Matthew McGill, Nicholas Roy, Avraham Ruderman, Daniel Tanis, and Frank Perbet for their assistance with research environment task development. We thank Alistair Muldal for assistance with data and infrastructure from prior efforts. We also thank Timothy Lillicrap for early input into the SIMA concept and insights from prior efforts. We thank Tom Ward, Joe Stanton, David Barker, and George Thomas for their infrastructure and support for running game binaries on Google Cloud infrastructure.

Finally, we thank our team of participants who generated gameplay and language annotation data, as well as performed human evaluations of our agents, without whom this work would not have been possible.

References

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

Josh Abramson, Arun Ahuja, Federico Carnevale, Petko Georgiev, Alex Goldin, Alden Hung, Jessica Landon, Jirka Lhotka, Timothy Lillicrap, Alistair Muldal, et al. Improving Multimodal Interactive Agents with Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2211.11602*, 2022a.

Josh Abramson, Arun Ahuja, Federico Carnevale, Petko Georgiev, Alex Goldin, Alden Hung, Jessica Landon, Timothy Lillicrap, Alistair Muldal, Blake Richards, et al. Evaluating Multimodal Interactive Agents. *arXiv preprint arXiv:2205.13274*, 2022b.

Adaptive Agent Team, Jakob Bauer, Kate Baumli, Satinder Baveja, Feryal Behbahani, Avishkar Bhoopchand, Nathalie Bradley-Schmieg, Michael Chang, Natalie Clay, Adrian Collister, et al. Human-Timescale Adaptation in an Open-Ended Task Space. In *International Conference on Machine Learning*, 2023.

Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional Foundation Models for Hierarchical Planning. In *Advances in Neural Information Processing Systems*, 2023.

Joshua Albrecht, Abraham Fetterman, Bryden Fogelman, Ellie Kitanidis, Bartosz Wróblewski, Nicole Seo, Michael Rosenthal, Maksis Knutins, Zack Polizzi, James Simon, et al. Avalon: A Benchmark

for RL Generalization Using Procedurally Generated Worlds. In *Advances in Neural Information Processing Systems*, 2022.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. PaLM 2 Technical Report. *arXiv preprint arXiv:2305.10403*, 2023.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. In *Advances in Neural Information Processing Systems*, 2022.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv preprint arXiv:1912.06680*, 2019.

Ioana Bica, Anastasija Ilić, Matthias Bauer, Goker Erdogan, Matko Bošnjak, Christos Kaplanis, Alexey A. Gritsenko, Matthias Minderer, Charles Blundell, Razvan Pascanu, and Jovana Mitrović. Improving fine-grained understanding in image-text pre-training. *arXiv preprint arXiv:2401.09865*, 2024.

Ivan Bratko, Tanja Urbančič, and Claude Sammut. Behavioural Cloning: Phenomena, Results and Problems. *IFAC Proceedings Volumes*, 28(21):143–149, 1995.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics Transformer for Real-World Control at Scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. *arXiv preprint arXiv:2307.15818*, 2023a.

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Conference on Robot Learning*, 2023b.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, 2020.

Cédric Colas, Tristan Karch, Nicolas Lair, Jean-Michel Dussoux, Clément Moulin-Frier, Peter Dominey, and Pierre-Yves Oudeyer. Language as a Cognitive Tool to Imagine Goals in Curiosity-Driven Exploration. In *Advances in Neural Information Processing Systems*, 2020.

Cédric Colas, Tristan Karch, Clément Moulin-Frier, and Pierre-Yves Oudeyer. Language and culture internalization for human-like autotelic AI. *Nature Machine Intelligence*, 4(12):1068–1076, 2022.

Erwin Coumans and Yunfei Bai. PyBullet, a Python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2023.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Association for Computational Linguistics*, 2019.

DeepMind Interactive Agents Team, Josh Abramson, Arun Ahuja, Arthur Brussee, Federico Carnevale, Mary Cassin, Felix Fischer, Petko Georgiev, Alex Goldin, Mansi Gupta, et al. Creating Multimodal Interactive Agents with Imitation and Self-Supervised Learning. *arXiv preprint arXiv:2112.03763*, 2021.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In *Advances in Neural Information Processing Systems*, 2022.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Conference on Robot Learning*, 2017.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. PaLM-E: An Embodied Multimodal Language Model. *arXiv preprint arXiv:2303.03378*, 2023.

Zane Durante, Bidipta Sarkar, Ran Gong, Rohan Taori, Yusuke Noda, Paul Tang, Ehsan Adeli, Shrinidhi Kowshika Lakshmikanth, Kevin Schulman, Arnold Milstein, et al. An Interactive Agent Foundation Model. *arXiv preprint arXiv:2402.05929*, 2024.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. In *International Conference on Machine Learning*, 2018.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. In *Advances in Neural Information Processing Systems*, 2022.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2312.11805*, 2023.

Karol Gregor, Danilo Jimenez Rezende, Frederic Besse, Yan Wu, Hamza Merzic, and Aaron van den Oord. Shaping Belief States with Generative Environment Models for RL. In *Advances in Neural Information Processing Systems*, 2019.

Caglar Gulcehre, Tom Le Paine, Bobak Shahriari, Misha Denil, Matt Hoffman, Hubert Soyer, Richard Tanburn, Steven Kapturowski, Neil Rabinowitz, Duncan Williams, et al. Making Efficient Use of Demonstrations to Solve Hard Exploration Problems. In *International Conference on Learning Representations*, 2019.

William H Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. MineRL: A Large-Scale Dataset of Minecraft Demonstrations. In *International Joint Conference on Artificial Intelligence*, 2019.

David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems*, 2018.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with Discrete World Models. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering Diverse Domains through World Models. *arXiv preprint arXiv:2301.04104*, 2023.

Stevan Harnad. The Symbol Grounding Problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990.

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. Grounded Language Learning in a Simulated 3D World. *arXiv preprint arXiv:1706.06551*, 2017.

Felix Hill, Andrew Lampinen, Rosalia Schneider, Stephen Clark, Matthew Botvinick, James L McClelland, and Adam Santoro. Environmental drivers of systematicity and generalization in a situated agent. In *International Conference on Learning Representations*, 2019.

Felix Hill, Olivier Tieleman, Tamara von Glehn, Nathaniel Wong, Hamza Merzic, and Stephen Clark. Grounded Language Learning Fast and Slow. In *International Conference on Learning Representations*, 2020.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Melissa Mozifian, Florian Golemo, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, et al. Sim2Real in Robotics and Automation: Applications and Challenges. *IEEE Transactions on Automation Science and Engineering*, 18(2): 398–400, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training Compute-Optimal Large Language Models. *arXiv preprint arXiv:2203.15556*, 2022.

Shengran Hu and Jeff Clune. Thought Cloning: Learning to Think while Acting by Imitating Human Thinking. *arXiv preprint arXiv:2306.00323*, 2023.

Yingdong Hu, Fanqi Lin, Tong Zhang, Li Yi, and Yang Gao. Look Before You Leap: Unveiling the Power of GPT-4V in Robotic Vision-Language Planning. *arXiv preprint arXiv:2311.17842*, 2023.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An Embodied Generalist Agent in 3D World. *arXiv preprint arXiv:2311.12871*, 2023.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents. In *International Conference on Machine Learning*, 2022.

Peter C Humphreys, David Raposo, Tobias Pohlen, Gregory Thornton, Rachita Chhaparia, Alistair Muldal, Josh Abramson, Petko Georgiev, Adam Santoro, and Timothy Lillicrap. A data-driven approach for learning to control computers. In *International Conference on Machine Learning*, 2022.

Yiding Jiang, Shixiang Shane Gu, Kevin P Murphy, and Chelsea Finn. Language as an Abstraction for Hierarchical Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2019.

Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. The Malmo Platform for Artificial Intelligence Experimentation. In *International Joint Conference on Artificial Intelligence*, 2016.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. Language Models can Solve Computer Tasks. In *Advances in Neural Information Processing Systems*, 2023.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks. *arXiv preprint arXiv:2401.13649*, 2024.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

Sreejan Kumar, Carlos G Correa, Ishita Dasgupta, Raja Marjieh, Michael Y Hu, Robert Hawkins, Jonathan D Cohen, Karthik Narasimhan, Tom Griffiths, et al. Using Natural Language and Program Abstractions to Instill Human Inductive Biases in Machines. In *Advances in Neural Information Processing Systems*, 2022.

Andrew K Lampinen, Nicholas Roy, Ishita Dasgupta, Stephanie CY Chan, Allison Tam, James Mcclelland, Chen Yan, Adam Santoro, Neil C Rabinowitz, Jane Wang, et al. Tell me why! Explanations support learning relational and causal structure. In *International Conference on Machine Learning*, 2022.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-Level Code Generation with AlphaCode. *Science*, 378(6624):1092–1097, 2022.

Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. STEVE-1: A Generative Model for Text-to-Behavior in Minecraft. *arXiv preprint arXiv:2306.00937*, 2023.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac Gym: High Performance GPU Based Physics Simulation For Robot Learning. In *Advances in Neural Information Processing Systems*, 2021.

James L McClelland, Felix Hill, Maja Rudolph, Jason Baldridge, and Hinrich Schütze. Placing language in an integrated understanding system: Next steps toward human-level performance in neural language models. *Proceedings of the National Academy of Sciences*, 117(42):25966–25974, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Hans Moravec. *Mind Children: The Future of Robot and Human Intelligence*. Harvard University Press, 1988.

Jesse Mu, Victor Zhong, Roberta Raileanu, Minqi Jiang, Noah Goodman, Tim Rocktäschel, and Edward Grefenstette. Improving Intrinsic Exploration with Language Abstractions. In *Advances in Neural Information Processing Systems*, 2022.

Kolby Nottingham, Prithviraj Ammanabrolu, Alane Suhr, Yejin Choi, Hannaneh Hajishirzi, Sameer Singh, and Roy Fox. Do embodied agents dream of pixelated sheep?: Embodied decision making using language guided world modelling. *arXiv preprint arXiv:2301.12050*, 2023.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, et al. Open-Ended Learning Leads to Generally Capable Agents. *arXiv preprint arXiv:2107.12808*, 2021.

OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. *arXiv preprint arXiv:2310.08864*, 2023.

Tim Pearce and Jun Zhu. Counter-Strike Deathmatch with Large-Scale Behavioural Cloning. In *IEEE Conference on Games*, 2022.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. VirtualHome: Simulating Household Activities via Programs. In *Computer Vision and Pattern Recognition*, 2018.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, Vladimír Vondruš, Theophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots. *arXiv preprint arXiv:2310.13724*, 2023.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A Generalist Agent. *Transactions on Machine Learning Research*, 2022.

Guillaume Sanchez, Honglu Fan, Alexander Spangher, Elad Levi, Pawan Sasanka Ammanamanchi, and Stella Biderman. Stay on topic with Classifier-Free Guidance. *arXiv preprint arXiv:2306.17806*, 2023.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A Platform for Embodied AI Research. In *International Conference on Computer Vision*, 2019.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *Computer Vision and Pattern Recognition*, 2020.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419): 1140–1144, 2018.

Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, Silvio Savarese, Hyowon Gweon, Jiajun Wu, and Li Fei-Fei. BEHAVIOR: Benchmark for Everyday Household Activities in Virtual, Interactive, and Ecological Environments. In *Conference in Robot Learning*, 2021.

Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Brianna Zitkovich, Fei Xia, Chelsea Finn, et al. Open-World Object Manipulation using Pre-trained Vision-Language Models. *arXiv preprint arXiv:2303.00905*, 2023.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training Home Assistants to Rearrange their Habitat. In *Advances in Neural Information Processing Systems*, 2021.

Allison Tam, Neil Rabinowitz, Andrew Lampinen, Nicholas A Roy, Stephanie Chan, DJ Strouse, Jane Wang, Andrea Banino, and Felix Hill. Semantic Exploration from Language Abstractions and Pretrained Representations. In *Advances in Neural Information Processing Systems*, 2022.

Weihao Tan, Ziluo Ding, Wentao Zhang, Boyu Li, Bohan Zhou, Junpeng Yue, Haochong Xia, Jiechuan Jiang, Longtao Zheng, Xinrun Xu, et al. Towards general computer control: A multimodal agent for red dead redemption ii as a case study. *arXiv preprint arXiv:2403.03186*, 2024.

Gerald Tesauro et al. Temporal Difference Learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Chen Tessler, Shahar Givony, Tom Zahavy, Daniel Mankowitz, and Shie Mannor. A Deep Hierarchical Approach to Lifelong Learning in Minecraft. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE International Conference on Intelligent Robots and Systems*, 2012.

Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. ChatGPT for Robotics: Design Principles and Model Abilities. *arXiv preprint arXiv:2306.17582*, 2023.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable Length Video Generation from Open Domain Textual Descriptions. In *International Conference on Learning Representations*, 2022.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint arXiv:2305.16291*, 2023a.

Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, et al. JARVIS-1: Open-World Multi-task Agents with Memory-Augmented Multimodal Language Models. *arXiv preprint arXiv:2311.05997*, 2023b.

Tom Ward, Andrew Bolt, Nik Hemmings, Simon Carter, Manuel Sanchez, Ricardo Barreira, Seb Noury, Keith Anderson, Jay Lemmon, Jonathan Coe, Piotr Trochim, Tom Handley, and Adrian Bolton. Using Unity to Help Solve Intelligence. *arXiv preprint arXiv:2011.09294*, 2020.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning Interactive Real-World Simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *Conference on Robot Learning*, 2020.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter Networks: Rearranging the Visual World for Robotic Manipulation. In *Conference on Robot Learning*, 2021.

Andy Zeng, Maria Attarian, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S Ryoo, Vikas Sindhwani, Johnny Lee, et al. Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language. In *International Conference on Learning Representations*, 2022.

Konrad Zolna, Serkan Cabi, Yutian Chen, Eric Lau, Claudio Fantacci, Jurgis Pasukonis, Jost Tobias Springenberg, and Sergio Gomez Colmenarejo. GATS: Gather-Attend-Scatter. *arXiv preprint arXiv:2401.08525*, 2024.

Author contributions

In this section, we summarize author contributions by project area, role in the area, and then alphabetically per role. A role key is provided at the end.

Agents & models

Leads:
Andrew Lampinen
Hubert Soyer

Partial Leads:
Danilo J. Rezende
Thomas Keck
Alexander Lerchner
Tim Scholtes

Past Leads:
Arun Ahuja
Ishita Dasgupta

Core Contributors:
Jeff Clune
Martin Engelcke
Ryan Faulkner
Karol Gregor
Rosemary Ke
Kavya Kopparapu
Yulan Liu
Joseph Marino
Hamza Merzic
Anna Mitenkova
Aneesh Pappu
John Reid
Daniel P. Sawyer
Daniel Slater
Heiko Strathmann
Allison Tam
Bojan Vujatovic
Zhengdong Wang

Contributors:
Stephanie Chan
Drew A. Hudson
Junkyung Kim
Loic Matthey
Pierre Harvey Richemond
Denis Teplyashin

Data

Leads:
Tayfun Terzi
Jane Wang

Core Contributors:
Junkyung Kim
Oscar Knagg
Renke Pan

Contributors:
Zhitao Gong
Andrew Lampinen
Anna Mitenkova
Yani Donchev
Davide Vercelli
John Reid

Environments: external

Leads:
Frederic Besse
Tim Harley
Piermaria Mendolicchio

Core Contributors:
Sarah Chakera
Vikki Copeman
Yani Donchev
Arne Olav Hallingstad
Maria Loks-Thompson
Tyson Roberts
Peter Stys

Contributors:
Charles Gbadamosi
Davide Vercelli
Duncan Williams

Environments: internal

Leads:

David Reichert

Past Leads:
Alex Cullum

Core Contributors:
Andrew Bolt
Bethanie Brownfield
Sarah Chakera
Dario de Cesare
Charles Gbadamosi

Mimi Jasarevic
Laura Kampis
Marjorie Limont
Piermaria Mendolicchio
Yanko Oliveira
Alex Platonov
Ollie Purkiss
Giles Ruscoe
Tasha Sandars
Guy Simmons
Nathaniel Wong
Nick Young

Contributors:
Catarina Barros
Gavin Buttimore
Adrian Collister
Julia Di Trapani
Emma Dunleavy
Sam Haves
Siobhan Mcloughlin
Valeria Oliveira
Haroon Qureshi
Davide Vercelli
Marcus Wainwright
Sarah York

Advisors:
Adrian Bolton
Max Cant

Evaluation

Leads:
Laura Kampis

Partial Leads:
Tim Harley
Andrew Lampinen

Core Contributors:
Martin Engelcke
Loic Matthey
Tim Scholtes
Daniel Slater
Davide Vercelli

Contributors:
Bethanie Brownfield
Sarah Chakera
Anna Mitenkova
David Reichert
John Reid
Jaume Sanchez Elias
Peter Stys
Jane Wang

Partnerships & legal

Leads:
Maria Abi Raad
Ed Hirst
Alexandre Moufarek

Core Contributors:
Kathryn Martin Cussons
Piermaria Mendolicchio

Project

Concept:
Frederic Besse
Tim Harley
Shane Legg

Project Leads:
Frederic Besse
Tim Harley
Hannah Openshaw

Past Project Leads:
Felix Hill
Shane Legg

Technical Leads:
Thomas Keck
Tayfun Terzi

Core Contributors:
Lucy Gonzales
Steph Hughes-Fitt

Product Manager:
Alexandre Moufarek

Advisors:
Jeff Clune
Daan Wierstra

Writing & design

Leads:
Andrew Lampinen
Joseph Marino

Core Contributors:
Martin Engelcke

Tim Harley
Laura Kampis
Yulan Liu
Daniel P. Sawyer
Jane Wang
Zhengdong Wang

Contributors:
Frederic Besse
Max Cant
Jeff Clune
Frankie Garcia
David Reichert

Role key
Lead: Responsible for the project area for the whole duration of the project.
Partial or Past Lead: Responsible for the project area for a part of the project duration.
Core Contributor: Contributed to the project area for an extended period of time.
Contributor: Contributed to the project area for a shorter period of time.
Advisor: Provided advice, feedback, and guidance to the project area.
Project Lead: Responsible for all aspects of the project for the whole duration of the project.
Past Project Lead: Responsible for all aspects of the project for a part of the project duration.
Technical Lead: Responsible for the technical direction of the project.