


Embodied large language models enable robots to complete complex tasks in unpredictable environments

Received: 22 June 2024

Accepted: 31 January 2025

Published online: 19 March 2025

 Check for updatesRuaridh Mon-Williams^{1,2,3}✉, Gen Li¹✉, Ran Long¹, Wenqian Du^{1,4} & Christopher G. Lucas¹

Completing complex tasks in unpredictable settings challenges robotic systems, requiring a step change in machine intelligence. Sensorimotor abilities are considered integral to human intelligence. Thus, biologically inspired machine intelligence might usefully combine artificial intelligence with robotic sensorimotor capabilities. Here we report an embodied large-language-model-enabled robot (ELLMER) framework, utilizing GPT-4 and a retrieval-augmented generation infrastructure, to enable robots to complete long-horizon tasks in unpredictable settings. The method extracts contextually relevant examples from a knowledge base, producing action plans that incorporate force and visual feedback and enabling adaptation to changing conditions. We tested ELLMER on a robot tasked with coffee making and plate decoration; these tasks consist of a sequence of sub-tasks from drawer opening to pouring, each benefiting from distinct feedback types and methods. We show that the ELLMER framework allows the robot to complete the tasks. This demonstration marks progress towards scalable, efficient and ‘intelligent robots’ able to complete complex tasks in uncertain environments.

If Deep Blue (the first computer to win a chess match against a reigning world champion) was truly intelligent, then should it not be able to move its own pieces when playing chess? Intelligence is a multifaceted construct and, thus, difficult to define. Consequently, human intelligence and its assessment is a controversial topic¹. However, there is a growing consensus that human intelligence is best understood as ‘embodied cognition’, where attention, language, learning, memory and perception are not abstract cognitive processes constrained to the brain but intrinsically linked with how the body interacts with its surrounding environment^{2,3}. Indeed, there is growing evidence that human intelligence has its ontological and phylogenetic foundations in sensorimotor processes⁴.

Embodied cognition has theoretical implications for ‘machine intelligence’ as it suggests that machines will be unable to demonstrate some aspects of intelligence if ‘cognitive’ processes are not

embedded in a robotic device. This is a conjecture that is still to be tested, but ‘intelligent robots’ provide an effective way of exploring various hypotheses concerning human intelligence and advancing the field of machine intelligence. More practically, effective human–robot collaboration will ultimately require robots to at least approximate ‘human-like’ capabilities. Thus, a reasonable expectation of future ‘intelligent machines’ is that they will have the potential to perform abstract cognitive computations as they skilfully interact with objects and humans within their environment⁵.

So far, parallel streams of activity have advanced: (1) the sensorimotor abilities of robots and (2) artificial intelligence⁶. We set out to test the hypothesis that these approaches can now be combined to create a step change in the ability of robots to show human-like intelligence. We further hypothesized that integrating (1) and (2) would allow robots to undertake the type of complex tasks that are practically useful in a

¹University of Edinburgh, Edinburgh, UK. ²Massachusetts Institute of Technology, Boston, MA, USA. ³Princeton University, Princeton, NJ, USA.

⁴Alan Turing Institute, London, UK. ✉e-mail: ruaridh.mw@ed.ac.uk; li.gen@ed.ac.uk

wide range of settings but currently outwith the capability of robotic systems. Consider a scenario in which someone returns home feeling fatigued and thirsty. A robot with a sophisticated manipulation system is situated in the homeowner's kitchen and is instructed to prepare a drink. The robot decides that a reinvigorating cup of coffee needs to be made and handed to their carbon companion. This task—straightforward for humans—encompasses a series of challenges that test the limits of current robotic capabilities^{7–11}. First, the robot must interpret the information it receives and analyse its surroundings. Next, it may need to search the environment to locate a mug. This could involve opening drawers with unspecified opening mechanisms. Then, the robot must measure and mix the precise ratio of water to coffee. This requires fine-grained force control and adaptation to uncertainty if, for example, the human moves the location of the mug unexpectedly^{9,12}. This scenario is a canonical example of the multifaceted nature of complex tasks in dynamic environments. Robotic systems have traditionally struggled with these tasks because they have been unable to follow high-level commands, have relied on preprogrammed responses and lack the flexibility to adapt seamlessly to perturbations^{13,14}.

Reinforcement learning and imitation learning have demonstrated the effectiveness of interaction and demonstration in teaching robots to perform complex tasks. These approaches are promising¹⁵, but often struggle with adaptation to novel tasks and coping with diverse scenarios. Imitation learning also faces challenges when a robot needs to adapt to new contexts^{16–23}. Nature-inspired machine intelligence provides a potential solution to these challenges. The sophistication of human manipulation is due, in part, to the type of cognitive processes that are captured artificially by large language models (LLMs)^{24–26}. LLMs offer a way to process complex instructions and adapt actions accordingly because of their advanced contextual understanding and generalization abilities^{27,28}.

A large body of recent research has used LLMs for short-horizon tasks^{15,27,29}. For instance, VoxPoser utilizes LLMs to perform a variety of everyday manipulation tasks¹⁵. Similarly, Robotics Transformer (RT-2) leverages large-scale web and robotic learning data, enabling robots to perform tasks beyond the training scenario with remarkable adaptability²⁹. Hierarchical diffusion policy introduces a model structure to generate context-aware motion trajectories, which enhances task-specific motions from high-level LLM decision inputs³⁰. However, challenges remain in effectively integrating LLMs into robotic manipulation. These challenges include complex prompting requirements, a lack of real-time interacting feedback, a dearth of LLM-driven work exploiting the use of force feedback and inefficient pipelines that hinder the seamless execution of tasks^{15,31}. Moreover, current approaches have neglected the application of retrieval-augmented generation (RAG)³² in robotics despite RAG's potential to continually update and refine robot knowledge with relevant and accurate examples (and increase the knowledge base without impacting performance). Robot capacity is also limited because force and visual feedback are typically not integrated in robot sensorimotor control^{15,33}. This integration is crucial in scenarios such as pouring water into a moving cup, where vision is necessary to track the cup and force feedback is needed for pouring the desired amount of water when vision is occluded^{16,34,35}. Thus, there is a need for an innovative approach in robot manipulation that combines the latest artificial 'cognition' with integrated 'sensorimotor' visual and force feedback capabilities to effectively execute actions in the face of uncertainty. Supplementary Section 1 provides more background on state-of-the-art approaches and their current limitations^{36–53}.

Embodied LLM-enabled robot (ELLMER) is a framework that integrates approaches in artificial intelligence and sensorimotor control to create a step change in robotic capabilities. Its usefulness arises from its combined use of vision and force for sensorimotor feedback control uniquely coupled with the cognitive capabilities afforded through an integrated LLM combined with RAG and a curated knowledge base. We hypothesized that ELLMER would allow a robot

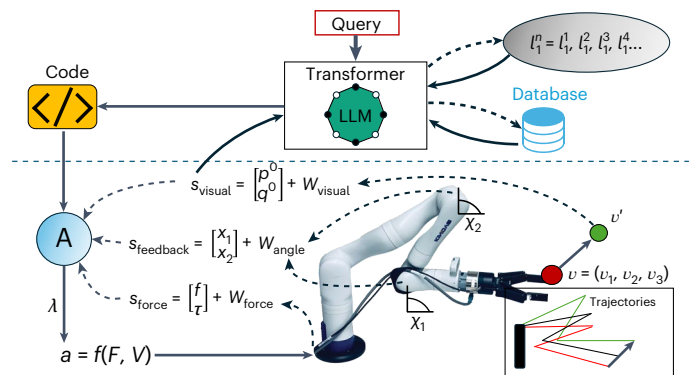


Fig. 1 | Schematic of the system framework. The schematic illustrates the system framework, showing the high-level (above the blue dashed horizontal line) and low-level (below the blue dashed horizontal line) system architecture. User queries are fed into a transformer via voice recognition software. The transformer (GPT-4) takes this input and integrates it with (i) an image (C) of the environment (via an azure Kinect depth camera); (ii) knowledge base of code examples, including various functions stored in a database. The transformer can decompose the higher-order abstracted task into actionable high-level sub-tasks, retrieve relevant code examples from the knowledge base, adapt them and write Python (v.3.8) code tailored to these tasks. The resultant code is then sent to the robot controller (A). The controller processes the code and sends control signals (λ) to the robot. The actions (a) are controlled with force (F) and vision (V) feedback. The model uses vision to identify the properties of different objects (for example, pose X of a coffee cup), so it can grasp objects accurately. The robot uses force (f) and torque (τ) feedback (available via an ATI force transducer) to manipulate objects skilfully (for example, determine how much water to pour). Feedback is necessary due to noise within the vision signal (η_{vision}), the robot joint angles (η_{angle}) and the force transducer signal (η_{force}). The feedback updates the motion in the ROS to achieve the desired goal through velocity commands of both linear (v_{xyz}) and angular (v_{rpy}) velocities. These commands generate trajectories based on appropriate forces and spatiotemporal patterns to achieve the sub-goals. The use of feedback loops, including 40-Hz updates of the end-effector position (p) and orientation (q), allow the robot to respond to disturbance (for example, the robot tracking a cup to determine its new position after it is moved by the user).

to make a cup of coffee for a human. We tested this hypothesis using a seven-degrees-of-freedom Kinova robotic arm to execute the complex, force-intensive task in an uncertain environment, leveraging integrated force and vision feedback. The overall system diagram is presented in Fig. 1.

Results

Language integration for abstract reasoning

We found that our approach allowed the robot to respond to an abstract high-order verbal prompt ('I'm tired, with friends due for cake soon. Can you make me a hot beverage, and decorate the plate with a random animal of your choice') and complete the desired task of making a hot drink and drawing a shape on a plate. The video demonstrating this interaction is provided in Supplementary Video 1. The robot responded to the prompt with the following sound output: 'I'm setting things up to make a hot beverage, and also to decorate the plate with a random animal drawing. First, I'll find a mug then scoop coffee and pour hot water. After that I'll take an item, likely a pen, to draw a random animal on the plate. It sounds like a fun and cosy time with cake soon.'. In particular, the robot chose to make a cup of coffee when it responded to the request for a hot beverage for a tired human, and was able to decompose the higher-order behaviour into a series of sub-tasks.

The integration of GPT-4 (ref. 54) was found to equip the robot with the desired capacity for abstract reasoning. GPT-4 is a language model that enables a robot to process user queries and environmental data to break down tasks into actionable steps. Our system was able

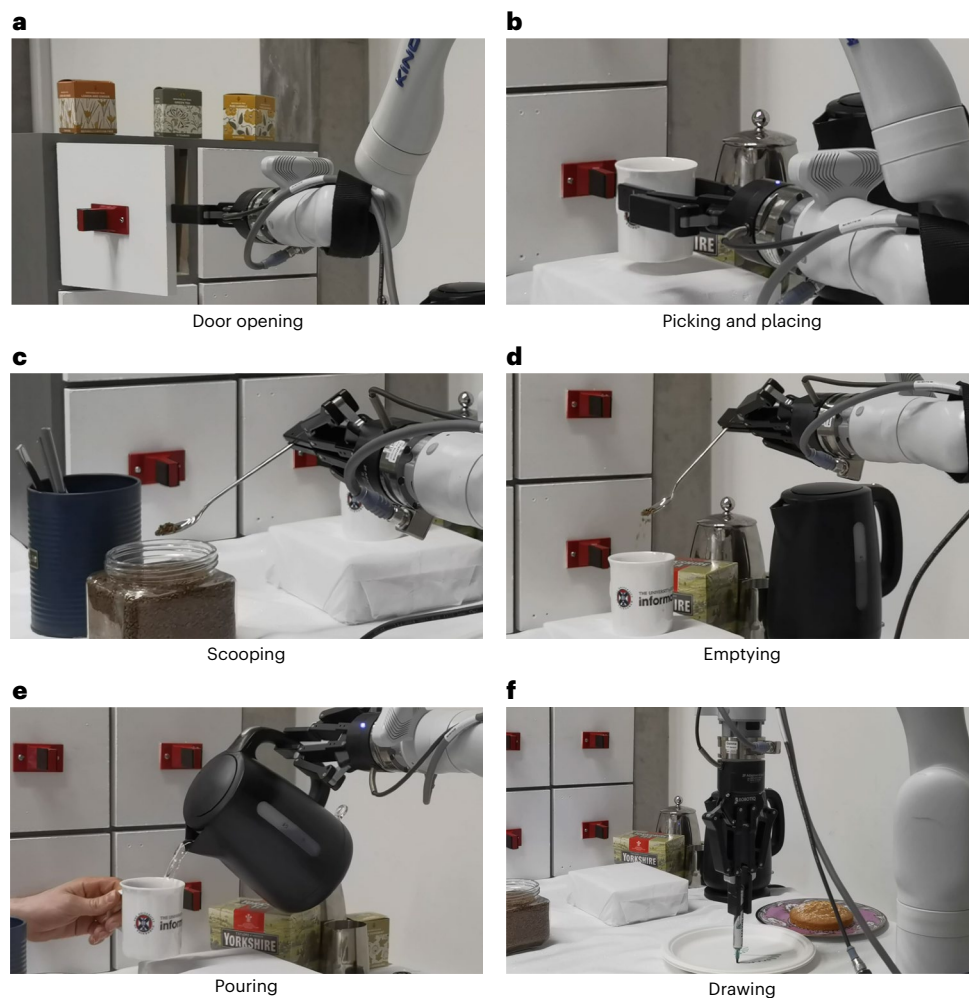


Fig. 2 | Kinova robot in action. a–f, Action shots of the Kinova Gen3 robot preparing coffee (a–e) and decorating a plate (f).

to generate code and execute actions with force and vision feedback, effectively providing the robot with a form of intelligence. Our methodology was successful in creating a custom GPT-4 (refs. 54,55) with a comprehensive database of flexible motion examples. The database successfully incorporated pouring, scooping, drawing, handovers, pick and place, and opening doors.

We found that the robot could identify and extract relevant examples for the downstream task using RAG. We explored various approaches to determine how intelligent machines could make the best use of RAG via our framework. These approaches included customizable open-source methods, such as Haystack⁵⁶ and Vebra³⁷, as well as proprietary technologies such as Azure Cloud AI. We found that all of these approaches were viable. In our experiment, we chose the simplest method: logically organizing our curated knowledge base in a markdown file and uploading it to the custom GPT API via the ‘Knowledge’ feature in the GPT’s platform. This allowed the platform to automatically handle the retrieval processes and select between semantic search (returning relevant text chunks) or document review (providing complete documents or sections from larger texts). We chose this solution as it provided a state-of-the-art embedder and model, gave ease of use and was able to consistently produce good performance in our task. However, our framework allows the incorporation of a range of RAG techniques and ensures that the ‘intelligent robot’ is able to efficiently complete complex tasks. The curated knowledge base, combined with RAG, allowed the language model to access a large selection of low- and high-order functions, each with

known uncertainties. Our tests showed that this capability enabled the robot to handle numerous scenarios effectively.

Completing a complex task

The robot was found to skilfully execute the high-level task specified by the user and was able to access a comprehensive motion primitive database. The database included a variety of flexible examples of specific motions and these were successfully carried out by the robotic arm (Fig. 2). Included in the database were examples of pouring liquids; scooping powders; opening doors with unknown mechanisms; picking up and placing objects; drawing any requested shape; conducting handovers; and moving in various directions, orientations or relative to specified objects. The robot was able to replicate and adapt the motions needed to execute the complex tasks requested by the user. The system enabled the robot to dynamically adjust to environmental variables and uncertainties. This enhanced the robot’s effectiveness in unpredictable conditions, and improved its flexibility and adaptability in the real-world setting.

Zero-shot pose detection

We found that an Azure Kinect DK Depth Camera, set to a resolution of 640×576 px² with a sample rate of 30 fps for depth sensing, was able to provide sufficient visual input for our method. We achieved calibration using a 14-cm AprilTag, and found that this allowed alignment between the camera and the robot’s base to an accuracy of less than 10^{-6} . This setup enabled accurate object position detection within

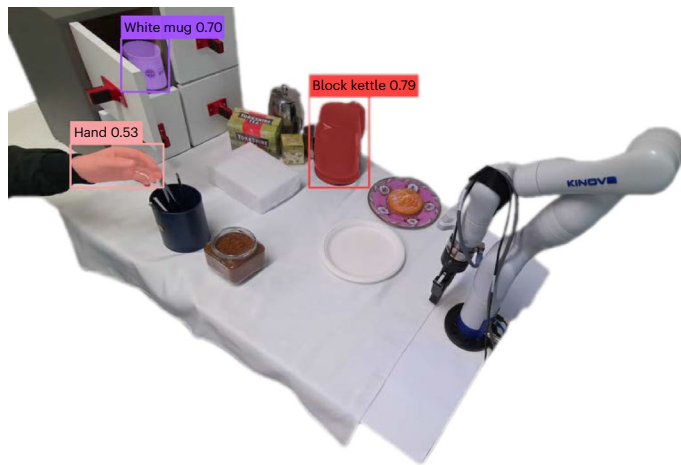


Fig. 3 | Vision detection module. Illustration of the zero-shot vision detection module identifying a hand, white mug and black kettle, as well as extracting target poses for robotic grasping.

the scene. Grounded-Segment-Anything⁵⁸ was successfully deployed for our language-to-vision module.

The vision system generated a three-dimensional (3D) Voxel representation that was effective at identifying object poses in our setup (the used Grounding DINO detection module achieved an average precision of 52.5 on the COCO zero-shot transfer benchmark). For example, we found that the module was able to correctly identify the white cup we used 100% of the time under our experimental conditions.

The 3D Voxel representation contained the meshes of various objects. From these meshes, target poses were extracted at a frequency of 1/3 Hz. In principle, the system should have been able to detect any object. In the pilot work, however, we established that the system would not always accurately identify the different objects associated with making hot beverages. This was often due to confusion between objects with similar shapes or objects absent from the training dataset. We also found that occlusion caused by the robot's end-effector could sometimes result in inaccuracies in object detection and lead to errors when we used highly cluttered environments. For example, the mean successful identification rate for a white cup was ~90% at occlusion ratios between 20% and 30%, but decreased substantially at higher occlusion ratios (for example, to ~20% for occlusion ratios between 80% and 90%). We anticipate that improvements in computer vision will enhance the ability of robots to deal with even the most visually complex environments. However, the performance of the vision system was impressive, and we found that our system could cope well with relatively unconstrained environments if the identified issues (for example, using out-of-distribution objects) were avoided (Fig. 3).

Force feedback

We found that an ATI multi-axis force and torque sensor provided sufficient force feedback for skilful object interaction. The sensor provided six components of force and torque, and the forces exerted by the robot's end-effector during task execution were successfully measured. We found that the sensor's accuracy was within ~2% of the full scale at a sampling rate of 100 Hz.

The robot was found to demonstrate a variety of motion dynamics accompanied by distinct types of force feedback during task execution. Figure 4 illustrates the forces experienced as the robot was preparing coffee and handing over a pen. As shown in Fig. 4, a diverse spectrum of external forces was handled across various tasks. For example, when putting down a mug, the peak upward force was used as an indicator of successful placement. By contrast, during drawer manipulation, the forces and torques along the *x* and *y* axes were critical, highlighting

their importance for successful task execution. The variability in force feedback exemplifies the advantages of our scalable approach that adapts to the requirements of diverse motions.

The pouring accuracy achieved was ~5.4 g per 100 g at a pitch velocity of 4 m s⁻¹. We assumed a quasi-static equilibrium to estimate the volume of water poured at any given moment. However, as the pitch velocity increased, the accuracy decreased, with errors approaching ~20 g s⁻¹ at a pitch velocity of 30 m s⁻¹. This decrease in accuracy can be attributed to the breakdown of the quasi-static assumption and the impact of the mass distribution of both pouring medium and container on measurement accuracy.

Generating art

DALL-E⁵⁹ was found to successfully produce an image from which we could derive a drawing trajectory. It was found that this enabled the robot to draw any design specified by the user. We found that DALL-E was able to create silhouettes based on keywords extracted from the user, such as 'random bird' or 'random plant'. The silhouette's outline was extracted and transformed to match the dimensions of the target surface. This allowed the robot to replicate the design on various physical objects (Fig. 5). We found that force feedback applied an even pen pressure when drawing, and this allowed control over the *z* component (Supplementary Section 2).

Evaluation

We evaluated our method for generating robotic plans against VoxPoser, which does not utilize RAG or force feedback. To compare the methods, we prompted an LLM to generate 80 human-like queries, reflecting the range of tasks specified in the knowledge base. These queries were then used to generate robot plans. We compared the performance outcomes from using RAG (our method)—in which the knowledge base is dynamically integrated into the LLM's decision-making process—to a baseline (VoxPoser) in which the knowledge base was statically incorporated into the LLM's context window. It is important to note that the second approach lacks scalability and becomes impractical as the knowledge base expands.

We evaluated the results based on answer faithfulness, which assesses an answer's truthfulness and accuracy (ensuring factual representation without fabrication or 'hallucination' errors). In our findings, using RAG improved the faithfulness of responses. For GPT-4 (gpt-4-0613), the faithfulness score increased from 0.74 to 0.88 with RAG. Similarly, GPT-3.5-turbo (gpt-3.5-turbo-0125) achieved 0.86 with RAG compared with 0.78 without it, and Zephyr-7B-beta saw an increase

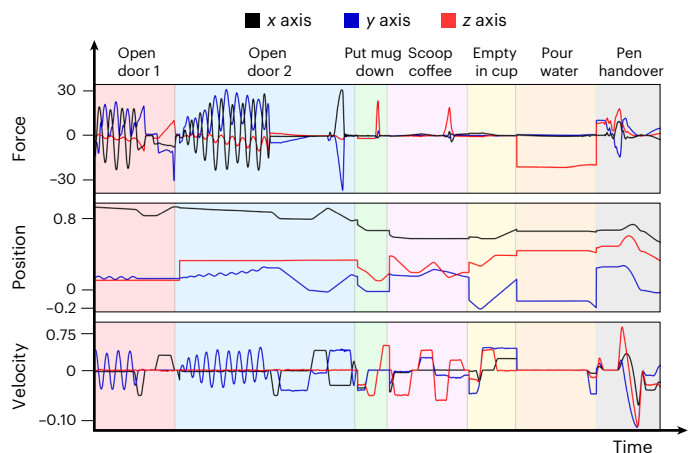


Fig. 4 | Force, velocity and position feedback. Force (N), velocity (m s⁻¹) and position (m) plots during the robot's coffee preparation, illustrating the diverse force feedback across the different motions. The drawing component has been omitted for clarity.

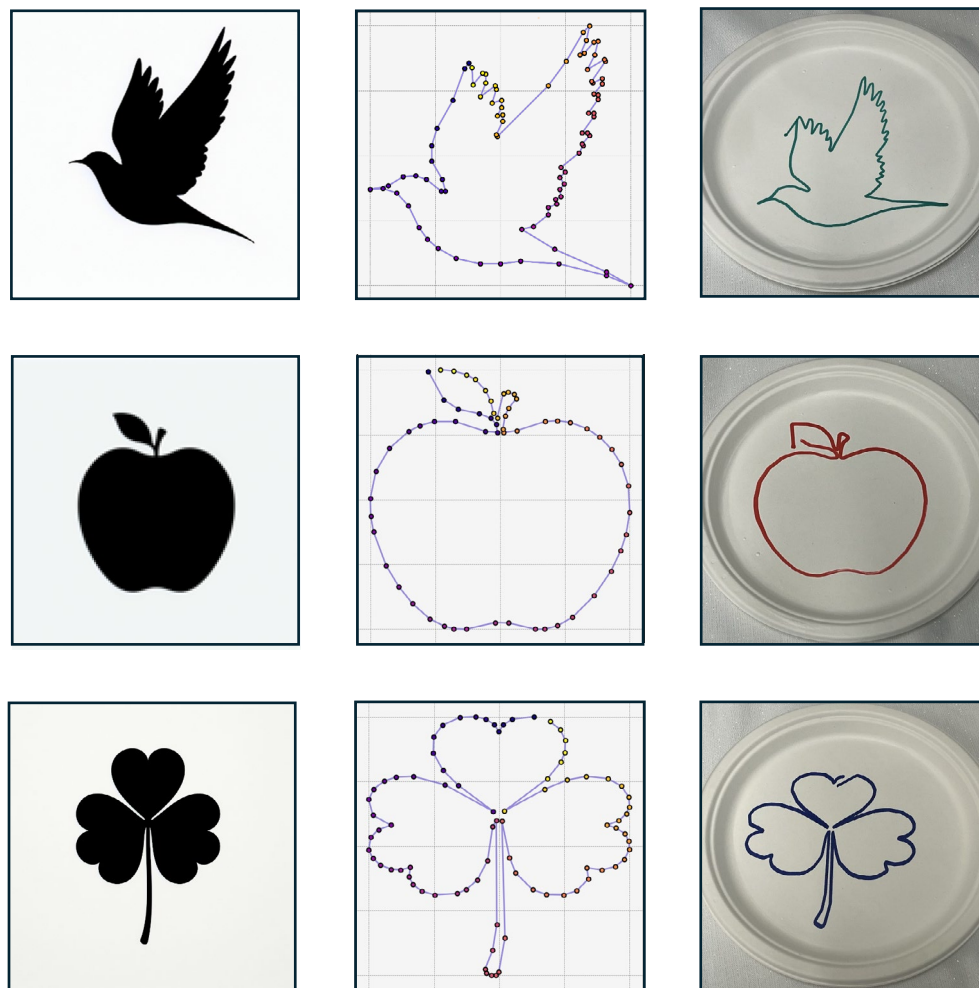


Fig. 5 | Drawing process visualization. Illustration of the drawing process across different queries. The top row shows the generated image, contour plot and drawing produced when instructed to create a ‘random animal’. The second row displays the corresponding outputs for a ‘random food’ and the third row illustrates the results for a ‘random plant’.

from 0.37 to 0.44. The improvement in faithfulness is particularly key for robotic applications, where accurate execution during physical interactions is essential.

Discussion

We tested our methodology—the ELLMER framework—that combines techniques from artificial intelligence and robot manipulation to create an intelligent robot. Our approach successfully combined the cognitive abilities of LLMs with the sensorimotor skills of robots, enabling our robot to interpret a high-order verbal command and execute a complex long-horizon task while adeptly managing uncertainties. We used the LLM, augmented with feedback loops and RAG, to write expressive code and facilitate the manipulation sub-tasks required by the robot to achieve the high-level goal (making a hot beverage). ELLMER allowed real-time adaptation to environmental changes and leveraged a repository of precise solutions via RAG. This ensured accurate task execution and broad adaptability³².

ELLMER encoded known constraints into the code examples (‘motion functions’) and enabled rapid accommodation to numerous uncertainties, such as fluctuating ingredient quantities or opening unknown drawers—capabilities that other methods lack without extensive additional training^{29,33,60,61}. The integration of vision, force and language modalities enhanced the manipulation performance. The force sensors improved task precision (for example, pouring a precise

and accurate amount of liquid when vision was occluded), whereas the vision system identified object positions and movements. The language capabilities enabled the system to produce feedback within the code, which is critical for adjusting to new tasks. The curated knowledge base improved the LLM’s performance by tailoring information retrieval to the specific task specifications, and this ensured high-quality contextually relevant outputs. A curated knowledge base is a pragmatic element that enhances controllability, accuracy and scalability. In this context, RAG can be seen as providing a cultural milieu of knowledge from which a robot can draw. In particular, this mirrors the ‘intelligence’ afforded to humans through the cultural transmission of knowledge. Thus, our work shows that integrating advanced language models and sensorimotor control strategies allows robots to leverage the exponential advancements in LLMs, enabling more sophisticated interactions. This will usher in the next age of automation with unprecedented levels of autonomy and precision, accentuating the need to manage these advancements safely⁶².

ELLMER’s potential extends to creating intricate and artistic movements. For instance, a model like DALL-E allows trajectories to be derived from visual inputs and opens new avenues for robotic trajectory generation. This method can be widely applied in tasks such as cake decoration or latte art. In future work, incorporating queries and images will enable novel trajectory generation, allowing for increased versatility. Moreover, recent LLM enhancements are



Fig. 6 | Coffee and plate decoration. Kinova Gen3 robot having prepared the coffee and decorated a plate.

set to notably improve the fluidity and effectiveness of human–robot interactions. Our examples of coffee making and plate decoration represent only a subset of the complex task types that a sophisticated robot might be required to undertake. ELLMER is conducive to being scaled up, so it includes a wide range of possible long-horizon tasks. Thus, ELLMER could incorporate a database of feedback loops or ‘learning-from-demonstration’ examples to facilitate a wide variety of complex robotic manipulations.

ELLMER is based on two assumptions concerning computer vision: (1) the vision module accurately identifies and classifies objects within the scene and (2) a comprehensive affordance map of the utensils is available. We endowed our model with prior knowledge of the kettle, spoon and door handle affordances, but recent work suggests that affordances can be learned with minimal data^{63,64}. Our focus was not on object detection, but we noted that detection response times hindered optimal performance. In addition, ELLMER could adjust to real-time changes but struggled with proactive adaptations (for example, task switching midway without prior programming). In future iterations, more frequent querying of the language model would allow the reassessment and modification of overall plans based on new inputs. We also note that there are still challenges that need to be addressed, such as the sophisticated modelling of complex force dynamics (for example, the forces on the end-effector as a function of the flow rate, container size and liquid viscosity) and the integration of spatial awareness tools (such as OctoMaps, a robotic library for a 3D occupancy map). Incorporating tactile sensors and using soft robotic techniques would improve the robot’s ability to apply appropriate forces without causing damage. ELLMER provides a flexible platform for incorporating these research developments, enabling robots to use ‘sensory’ feedback to interpret material properties and precisely tailor the forces they apply.

The current iteration of ELLMER allowed the robot to successfully complete a complex task in ‘one shot’. This provides a compelling picture of the capabilities of intelligent machines that combine sensorimotor capabilities with the abstract reasoning provided by LLMs. Nevertheless, we anticipate that the robot capacity will increase exponentially as the components combined within ELLMER become ever more refined. Our framework is hardware agnostic and can be easily customized with open-source RAG solutions like Haystack, supporting quick adjustments to embedders, retrievers, chunking techniques and LLMs. ELLMER offers a flexible framework for researchers to collaboratively develop intelligent machines. Supplementary Section 3 provides more information on ELLMER and future research.

The power of our approach lies in the embodiment of cognition through a framework that combines enhanced sensorimotor abilities with the cognitive reasoning capabilities of LLMs. Through this combination, ELLMER enables robots to explore and interact with their environment more effectively, emulating aspects of the connection between

experience and action observed in human intelligence. This opens up opportunities for robots to gain a form of ‘physical intelligence’, where their exploration of the environment drives the sensorimotor learning process. In conclusion, ELLMER integrates language processing, RAG, force and vision to enable robots to adapt to complex tasks. It combines the following features: (1) interpreting high-level human commands, (2) completing long-horizon tasks and (3) utilizing integrated force and vision signals to manage noise and disturbances in changing environments. ELLMER allows methods such as reinforcement learning, imitation learning and flexible motion primitives to be combined holistically for enhanced adaptability and ‘robot intelligence’ in diverse and dynamic scenarios. It demonstrates that integrating the cognitive reasoning capabilities of LLMs with robots’ sensorimotor skills allows them to interpret and manipulate their environment and complete complex tasks through embodied machine intelligence.

Methods

Overview

The goal of the robot was to respond to high-level human commands in a dynamic environment, such as a home kitchen. We designed a realistic setting featuring items including a kettle, white mug, drawers, kitchen paraphernalia and a coffee pot. The scenario was designed to test the robot’s ability to perform diverse tasks in a realistic, although reasonably constrained, environment as it interacts with a human present. We assumed that robotic low-level control mechanisms managed obstacle avoidance. The pipeline consisted of a language-processing component for task execution, a vision system for pose detection and a force module for object manipulation. All of this was integrated within a robotic operating system (ROS) process.

Specifically, our approach built on the ‘code for dynamic policies’ approach⁶⁵ that can facilitate adaptable robotic actions. In our implementation, we utilized GPT-4 and OpenAI’s RAG infrastructure. We leveraged LLMs’ capabilities using RAG³² to dynamically select and adapt the most suitable policy from a database or generate its own code based on relevant examples. In contrast to existing pure LLM-driven methods^{25,27,29}, we integrated force and vision into the framework, allowing the system to adapt to a variety of complex tasks in dynamic settings. This approach equips the robotic system with the capacity for high-level contextual understanding²⁵ and the proficiency to execute complex tasks with real-time feedback, ensuring accuracy and precision. The approach ensures that each action is aligned with the specific demands of the task and the environmental conditions (Fig. 6).

Hardware and software

A Kinova seven-degrees-of-freedom robot was used. An Azure Kinect Sensor was used at a resolution of 640×576 px² and 30 fps, along with an ATI multi-axis force sensor. A 140-mm Robotiq gripper was attached to the end of the robot. The force sensor was attached to the Robotiq gripper and Kinova arm using a 3D printed flange. A small cylinder was placed on the force sensor on the side closest to the gripper so that the movements of the gripper would not touch the force sensor, leading to readings being inaccurate. A Dell desktop computer with an Intel Core i9 processor with an NVIDIA RTX 2080 graphics-processing unit was used and connected to the robot with an Ethernet cable. Similarly, both Azure cameras were attached to the desktop. Ubuntu 20.04 and the ROS were used. Our code relied on the Kinova ROS Kortex library. The NVIDIA RTX 2080 utilizes ~225 W under typical load conditions⁶⁶, whereas the Kinova robotic arm consumes ~36 W (ref. 67). In our scenarios, each task runs for up to 4 min. Utilizing the EPA’s average conversion factor of ~0.4 kg of CO₂ per kWh for mixed energy sources⁶⁸, the carbon emission for each task comes to ~0.007 kg (7 g) of CO₂.

Language processing

The LLM processes an image and the user’s query, systematically breaking down the complex task L_i into a sequence of steps $\{L_1, L_2, \dots, L_N\}$,

where each step L_i may depend on the completion of the preceding steps. The sequence of steps is critical, and dependencies exist between steps; for example, if an object (for example, a mug) is required but not found, then potentially a cupboard should be opened.

The environmental data gathered from the initial image input are key in decomposing the abstract task. For instance, when asked to make a beverage, the ingredients present in the environment are critical in deciding which drink to make, and the visual information can help identify possible locations. The interface was facilitated by GPT-4, which ran under the instruction to write and dispatch code to a robot via the server platform. The process was assisted by a knowledge base containing code examples and allowed continuous communication with the robot. The curated knowledge base contained validated examples of low- and high-order actions that incorporate known uncertainties. Including these motion examples is key to enabling the robot to handle numerous scenarios and complete long-horizon tasks. High-level motion primitives or policies can compress multiple known uncertainties into a single function, reducing the need for extensive code writing. RAG allowed the knowledge base to be comprehensive without sacrificing performance. The system interacted with the ROS and communicated via a low-latency connection provided by the EC2 server through JSON action queries and responses.

The dependency among tasks is expressed through conditional probabilities such as $P(L_{2A}, L_{2B}|L_1)$, which specifies the likelihood of progressing to tasks L_{2A} or L_{2B} following the successful execution of task L_1 . This helps in planning the sequence of steps, ensuring the robot can adapt its actions based on real-time feedback. The LLM generates executable code that is sent to the server, based on the instructions (prompt) and a knowledge base containing examples. The code is run on the ROS in a secure environment that only has access to predefined functions, thereby ensuring safety in task execution.

RAG

A key feature of our system is the deployment of RAG. RAG integrates user queries with information from a continually updated, curated knowledge base, optimizing the output of the LLM. This approach allows the model to follow code examples provided in the database, ensuring accuracy, reliability and scalability as the knowledge base evolves.

We used vector RAG, which involves using an encoder to embed the query (q) and segments of the knowledge base ($\{s_1, s_2, \dots, s_m\}$), known as chunks, into vector representations. Chunks were then compared with the query based on cosine similarity, and the top k chunks were selected as contextually relevant information for generating responses. Alternative retrieval techniques that can be used within our framework include traditional RAG (keyword-/rule-based RAG) or hybrid retrieval methods.

The RAG pipeline can be customized by selecting different document stores (the medium in which the knowledge base is stored and organized). In our experimental test, we used the inbuilt OpenAI RAG process and organized our curated knowledge base in a markdown file as the document store. However, a range of other RAG approaches can be used in our framework, utilizing tools like Haystack⁵⁶ and Vebra⁵⁷. These tools allow users to select a range of document stores—from ‘markdown files’ for simple text-based knowledge to ‘Elasticsearch’ for complex, indexed data—along with specific embedders, retrievers and chunking techniques, as well as the LLM itself.

Vision system

Grounded-Segment-Anything was used as the language-to-vision model to create a 3D voxel that highlighted the positions of all objects and allowed their poses to be extracted for robotic grasping^{58,69}. This enabled (1) the generation of object-specific bounding boxes, (2) the manufacture of segmented masks via MobileSAM and (3) the creation of voxels that encapsulate the detected objects. The voxels allowed target object poses to be extracted.

Force module

To ensure accurate measurements in force-rich applications, we calibrated the ATI force sensor to compensate for gravitational forces, ensuring it registered zero in the absence of external forces. This calibration is key for accurately predicting the external forces applied to the end-effector. The process involved sequentially zeroing the force sensor on one axis, rotating the sensor and then zeroing on the next axis. The local forces were transformed into the global plane to estimate the upward force at different rotations $F_{\text{global}} = T_{\text{end-effector to robot base}} \times F_{\text{local}}$, where F_{global} is the force vector in the global (robot base) coordinate frame, $T_{\text{end-effector to robot base}}$ is the transformation matrix from the end-effector's frame to the robot's base frame and F_{local} is the force vector in the local coordinate frame of the end-effector. We explored various methods, such as moving the sensor's position and orientation and using polynomial functions for calibration. However, the simpler calibration method was found to be the most effective.

To estimate the flow rates, we assumed a condition of static equilibrium and maintain slow operational speeds during pouring. Mathematically, this is represented as $F_{\text{up}} \approx mg$ and $\Delta F_{\text{up}} \approx \Delta mg$. In situations involving variable acceleration, the relationship between forces and flow rates becomes more complex. It necessitates a dynamic model that accounts for varying inputs, such as flow rates, container's centre of mass and inertia of the end-effector, to map dynamic force inputs to the pouring flow rates.

The system continuously manages force vectors along three axes, adjusting the applied force based on the criteria within its knowledge base. The LLM dynamically selects the necessary force magnitudes and directions tailored to meet specific downstream task requirements. For example, the knowledge base may specify varying force magnitudes to be applied depending on the object characteristics or task demands. This approach enables the system to adjust its actions autonomously to align with a broad range of operational criteria.

ROS operation

In this work, we initiated the robotic processes by launching a Kinova ROS Kortex driver. This established a node that enables communication within the ROS network and the Kinova Gen3 robot. The node publishes several topics that subscribers can access, and it provides services that can be called to modify the robot's configuration. The base joints are updated at a frequency of 40 Hz. Concurrently, the Robotiq 2F-140 mm gripper node is activated at 50 Hz. The node sets up a communication link with the gripper via a USB connection, and it initiates an action server that enables precise control of the gripper and facilitates the exchange of operating data.

A vital element of our robotic system is the vision module node. A ‘classes’ variable is used to identify the target pose of selected objects within the environment. This variable can be dynamically updated, thereby allowing the system to adapt to changes in the scene. The pose coordinates of the objects, as established by the ‘classes’ variable, are published approximately at every $\sim \frac{1}{3}$ Hz. This is largely due to the processing time of Grounding DINO in detecting objects and establishing the bounding boxes. Moreover, we used an AprilTag to determine the position of the camera relative to the robot's base. This is represented as $P^R = T_{AR} \times (T_{CA} \times P^C)$, where P^C is the point in the camera frame, T_{CA} is the transformation matrix from the camera frame to the AprilTag, T_{AR} is the transformation matrix from the AprilTag to the robot's base and PR is the point in the robot's base frame.

In parallel, a force node is launched at a frequency of 100 Hz and provides multi-axis force and torque readings, localized to the ATI force transducer. The readings are transformed using a quaternion-based 3×3 rotation matrix to align with the global base frame of the robot, providing raw and averaged values over the last five time steps across fixed degrees of freedom. It calculates forces in the global frame of the robot base using the rotational matrix, calculated from the kinematic data.

ROS facilitates the continuous processing of multimodal feedback data from the language processing, vision systems, force metrics and joint end-effector positions. The motions operate on a foundational six-degrees-of-freedom twist command, which controls velocity and the variable speed and force gripper procedures for opening and closing. This enables the integration of hard-coded safety constraints, such as maximum velocity and force limits, as well as workspace boundaries.

The linear velocities were clamped within $\pm 0.05 \text{ m s}^{-1}$ and the angular velocities were clamped within $\pm 60^\circ \text{ s}^{-1}$. End-effector forces were also limited to 20 N. This is coded into the fundamental motion primitives; therefore, error in the language model will not override this. The end-effector is also clamped within the predefined workspace bounds of $x = [0.0, 1.1]$, $y = [-0.3, 0.3]$ and $z = [0, 1.0]$. This is checked in future time steps by a publisher at a frequency of 10 Hz.

Data availability

The dataset used in this work is available in an open-source GitHub repository at <https://github.com/ruaridhmon/ELLMER>.

Code availability

The code supporting this study is available via GitHub at <https://github.com/ruaridhmon/ELLMER> and has been archived in Zenodo at <https://doi.org/10.5281/zenodo.14483539> (ref. 70).

References

- Intelligence research should not be held back by its past. *Nature* **545**, 385–386 (2017).
- Friston, K. Embodied inference and spatial cognition. *Cogn. Process.* **13**, 497–514 (2012).
- Wilson, M. Six views of embodied cognition. *Psychon. Bull. Rev.* **9**, 625–636 (2002).
- Clark, A. An embodied cognitive science. *Trends Cogn. Sci.* **3**, 345–351 (1999).
- Stella, F., Della Santina, C. & Hughes, J. How can LLMs transform the robotic design process? *Nat. Mach. Intell.* **5**, 561–564 (2023).
- Miriyev, A. & Kovac, M. Skills for physical artificial intelligence. *Nat. Mach. Intell.* **2**, 658–660 (2020).
- Cui, J. & Trinkle, J. Toward next-generation learned robot manipulation. *Sci. Robot.* **6**, eabd9461 (2021).
- Arents, J. & Greitans, M. Smart industrial robot control trends, challenges and opportunities within manufacturing. *Appl. Sci.* **12**, 937 (2022).
- Billard, A. & Kragic, D. Trends and challenges in robot manipulation. *Science* **364**, eaat8414 (2019).
- Yang, G.-Z. et al. The grand challenges of *Science Robotics*. *Sci. Robot.* **3**, eaar7650 (2018).
- Buchanan, R., Rofer, A., Moura, J., Valada, A. & Vijayakumar, S. Online estimation of articulated objects with factor graphs using vision and proprioceptive sensing. In *2024 IEEE International Conference on Robotics and Automation (ICRA)* 16111–16117 (IEEE, 2024).
- Nikolaidis, S., Ramakrishnan, R., Gu, K. & Shah, J. Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* 189–196 (IEEE, 2015).
- Saveriano, M., Abu-Dakka, F. J., Kramberger, A. & Peternel, L. Dynamic movement primitives in robotics: a tutorial survey. *Int. J. Robot. Res.* **42**, 1133–1184 (2023).
- Kober, J. et al. Movement templates for learning of hitting and batting. In *2010 IEEE International Conference on Robotics and Automation* 853–858 (IEEE, 2010).
- Huang, W. et al. VoxPoser: composable 3D value maps for robotic manipulation with language models. In *Proc. 7th Conference on Robot Learning* 540–562 (PMLR, 2023).
- Zhang, D. et al. Explainable hierarchical imitation learning for robotic drink pouring. In *IEEE Transactions on Automation Science and Engineering* 3871–3887 (2022).
- Hussein, A., Gaber, M. M., Elyan, E. & Jayne, C. Imitation learning: a survey of learning methods. *ACM Comput. Surv.* **50**, 21:1–21:35 (2017).
- Di Palo, N. & Johns, E. DINOBot: robot manipulation via retrieval and alignment with vision foundation models. In *International Conference on Robotics and Automation (ICRA)* 2798–805 (IEEE, 2024).
- Shridhar, M., Manuelli, L. & Fox, D. CLIPort: what and where pathways for robotic manipulation. In *Proc. 5th Conference on Robot Learning* 894–906 (PMLR, 2022).
- Shridhar, M., Manuelli, L. & Fox, D. Perceiver-Actor: a multi-task transformer for robotic manipulation. In *Proc. 6th Conference on Robot Learning* 785–799 (PMLR, 2023).
- Mees, O., Hermann, L. & Burgard, W. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robot. Autom. Lett.* **7**, 11205–11212 (2022).
- Mees, O., Borja-Diaz, J. & Burgard, W. Grounding language with visual affordances over unstructured data. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* 11576–11582 (IEEE, 2023).
- Shao, L., Migimatsu, T., Zhang, Q., Yang, K. & Bohg, J. Concept2Robot: learning manipulation concepts from instructions and human demonstrations. *Int. J. Robot. Res.* **40**, 1419–1434 (2021).
- Ichter, B. et al. Do as I can, not as I say: grounding language in robotic affordances. In *Proc. 6th Conference on Robot Learning* 287–318 (PMLR, 2023).
- Driess, D. et al. PaLM-E: an embodied multimodal language model. In *Proc. 40th International Conference on Machine Learning* 8469–8488 (PMLR, 2023).
- Peng, A. et al. Preference-conditioned language-guided abstraction. In *Proc. 2024 ACM/IEEE International Conference on Human-Robot Interaction, HRI '24* 572–581 (Association for Computing Machinery, 2024).
- Huang, W., Abbeel, P., Pathak, D. & Mordatch, I. Language models as zero-shot planners: extracting actionable knowledge for embodied agents. In *Proc. 39th International Conference on Machine Learning* 9118–9147 (PMLR, 2022).
- Huang, J. & Chang, K. C.-C. Towards reasoning in large language models: a survey. In *Findings of the Association for Computational Linguistics: ACL 2023* 1049–1065 (Association for Computational Linguistics, 2023).
- Zitkovich, B. et al. RT-2: vision-language-action models transfer web knowledge to robotic control. In *Proc. 7th Conference on Robot Learning* 2165–2183 (PMLR, 2023).
- Ma, X., Patidar, S., Haughton, I. & James, S. Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 18081–18090 (IEEE, 2024).
- Zhang, C., Chen, J., Li, J., Peng, Y. & Mao, Z. Large language models for human-robot interaction: a review. *Biomimetic Intell. Robot.* **3**, 100131 (2023).
- Lewis, P. et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems* 9459–9474 (Curran Associates, 2020).
- Raiaan, M. et al. A review on large language models: architectures, applications, taxonomies, open issues and challenges. *IEEE Access* **12**, 26839–26874 (2024).
- Rozo, L., Jimenez, P. & Torras, C. Force-based robot learning of pouring skills using parametric hidden Markov models. In *9th International Workshop on Robot Motion and Control* 227–232 (IEEE, 2013).

35. Huang, Y., Wilches, J. & Sun, Y. Robot gaining accurate pouring skills through self-supervised learning and generalization. *Robot. Auton. Syst.* **136**, 103692 (2021).
36. Mon-Williams, R., Stouraitis, T. & Vijayakumar, S. A behavioural transformer for effective collaboration between a robot and a non-stationary human. In *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)* 1150–1157 (IEEE, 2023).
37. Belkhale, S., Cui, Y. & Sadigh, D. Data quality in imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)* 80375–80395 (Curran Associates, 2024).
38. Khazatsky, A. et al. DROID: a large-scale in-the-wild robot manipulation dataset. *Robotics: Science and Systems*; <https://www.roboticsproceedings.org/rss20/p120.pdf> (2024).
39. Acosta, B., Yang, W. & Posa, M. Validating robotics simulators on real-world impacts. *IEEE Robot. Autom. Lett.* **7**, 6471–6478 (2022).
40. Alomar, A. et al. CausalSim: a causal framework for unbiased trace-driven simulation. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)* 1115–1147 (USENIX Association, 2023).
41. Choi, H. et al. On the use of simulation in robotics: opportunities, challenges, and suggestions for moving forward. *Proc. Natl Acad. Sci. USA* **118**, e190785611 (2021).
42. Del Aguila Ferrandis, J., Moura, J. & Vijayakumar, S. Nonprehensile planar manipulation through reinforcement learning with multimodal categorical exploration. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 5606–5613 (IEEE, 2023).
43. Kirk, R., Zhang, A., Grefenstette, E. & Rocktäschel, T. A survey of zero-shot generalisation in deep reinforcement learning. *J. Artif. Intell. Res.* **76**, 201–264 (2023).
44. Dai, T. et al. Analysing deep reinforcement learning agents trained with domain randomisation. *Neurocomputing* **493**, 143–165 (2022).
45. Chang, J., Uehara, M., Sreenivas, D., Kidambi, R. & Sun, W. Mitigating covariate shift in imitation learning via offline data with partial coverage. In *Advances in Neural Information Processing Systems* 965–979 (Curran Associates, 2021).
46. Huang, W. et al. Inner monologue: embodied reasoning through planning with language models. In *Proc. 6th Conference on Robot Learning* 1769–1782 (PMLR, 2023).
47. Nair, S., Rajeswaran, A., Kumar, V., Finn, C. & Gupta, A. R3M: a universal visual representation for robot manipulation. In *Proc. 6th Conference on Robot Learning* Vol. 205, 892–909 (PMLR, 2022).
48. Singh, I. et al. ProgPrompt: generating situated robot task plans using large language models. In *Proc. IEEE/CVF International Conference on Robotics and Automation (ICRA)* 11523–11530 (IEEE, 2023).
49. Song, C. H. et al. LLM-Planner: few-shot grounded planning for embodied agents with large language models. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)* 2998–3009 (IEEE/CVF, 2023).
50. Vemprala, S. H., Bonatti, R., Bucker, A. & Kapoor, A. ChatGPT for robotics: design principles and model abilities. *IEEE Access* **12**, 55682–55696 (2024).
51. Ding, Y., Zhang, X., Paxton, C. & Zhang, S. Task and motion planning with large language models for object rearrangement. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 2086–2092 (IEEE, 2023).
52. Kwon, M. et al. Toward grounded commonsense reasoning. In *Proc. International Conference on Robotics and Automation (ICRA)* 5463–5470 (IEEE, 2024).
53. Hong, J., Levine, S. & Dragan, A. Learning to influence human behavior with offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)* 36094–36105 (Curran Associates, 2024).
54. OpenAI. GPT-4 technical report. Preprint at <http://arxiv.org/abs/2303.08774> (2024).
55. OpenAI. Custom models program: fine-tuning GPT-4 for specific domains (2023); <https://platform.openai.com/docs/guides/fine-tuning/>
56. Pietsch, M. et al. Haystack: the end-to-end nlp framework for pragmatic builders. *GitHub* <https://github.com/deepset-ai/haystack> (2019).
57. Weaviate. Verba: the golden RAGriever. *GitHub* <https://github.com/weaviate/Verba> (2023).
58. Kirillov, A. et al. Segment anything. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)* 4015–4026 (IEEE, 2023).
59. Ramesh, A. et al. Zero-shot text-to-image generation. In *Proc. 38th International Conference on Machine Learning* 8821–8831 (PMLR, 2021).
60. Zeng, A. et al. Socratic models: composing zero-shot multimodal reasoning with language. In *Proc. International Conference on Learning Representations (ICLR)* (2023).
61. Cui, Y. et al. No, to the right: online language corrections for robotic manipulation via shared autonomy. In *Proc. 2023 ACM/IEEE International Conference on Human-Robot Interaction, HRI '23* 93–101 (Association for Computing Machinery, 2023).
62. Bengio, Y. et al. Managing extreme AI risks amid rapid progress. *Science* **384**, 842–845 (2024).
63. Li, G., Jampani, V., Sun, D. & Sevilla-Lara, L. Locate: localize and transfer object parts for weakly supervised affordance grounding. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* 10922–10931 (IEEE, 2023).
64. Li, G., Sun, D., Sevilla-Lara, L. & Jampani, V. One-shot open affordance learning with foundation models. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* 3086–3096 (IEEE, 2024).
65. Liang, J. et al. Code as policies: language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* 9493–9500 (IEEE, 2023).
66. Hong, S. & Kim, H. An integrated GPU power and performance model. In *Proc. 37th Annual International Symposium on Computer Architecture* 280–289 (Association for Computing Machinery, 2010).
67. Kinova Robotics. Kinova Gen3 Ultra-Lightweight Robotic Arm User Guide (2023); <https://assets.iqr-robot.com/wp-content/uploads/2023/08/20230814163651088831.pdf>
68. US Environmental Protection Agency. GHG emission factors hub (2024); <https://www.epa.gov/climateleadership/ghg-emission-factors-hub>
69. Liu, S. et al. Grounding DINO: marrying DINO with grounded pre-training for open-set object detection. In *2024 European Conference on Computer Vision* (eds Leonidis, A. et al.) Vol. 15105 (Springer, 2023).
70. ruaridhmon. ruaridhmon/ELLMER: v1.0.0: Initial Release. *Zenodo* <https://doi.org/10.5281/zenodo.14483539> (2024).

Acknowledgements

This work was supported by the EPSRC CDT in RAS (EP/L016834/1). We thank S. Vijayakumar for his support and for providing access to resources; L. Martins and the Edinburgh Workshop for their assistance with hardware; and J. Wang, T. Stouraitis, J. Ferrandis and many others for their invaluable support and expertise.

Author contributions

Conceptualization: R.M.-W., G.L. and R.L. Methodology: R.M.-W., G.L., R.L., W.D. and C.G.L. Software: R.M.-W., G.L., R.L. and W.D. Formal analysis: R.M.-W. Investigation: R.M.-W. and W.D. Visualization: R.M.-W. Validation: R.M.-W. and W.D. Writing—original draft:

R.M.-W. Writing—review and editing: R.M.-W., G.L., R.L. and C.G.L.
Supervision: C.G.L.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s42256-025-01005-x>.

Correspondence and requests for materials should be addressed to Ruaridh Mon-Williams or Gen Li.

Peer review information *Nature Machine Intelligence* thanks Matteo Saveriano and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2025