Ludi$_{0.1}$: An Agentic System for Socially Intelligent Robots

Ludo Robotics[†]

Robot foundation models have substantially advanced perception and control, but natural human–robot collaboration requires more than executing isolated commands. A robot must recognize ambiguity, maintain context across turns, communicate its intentions, and revise ongoing behavior as the user's intent changes. We present **Ludi$_{0.1}$**, an agentic system for socially intelligent robots that integrates interactive speech, multimodal reasoning, memory, navigation, and learned manipulation. Its decision-making core is a fine-tuned vision–language model trained on multi-turn interaction traces spanning ambiguous requests, clarifications, corrections, interruptions, mixed social and task dialogue, and multi-step tasks. A purpose-built harness manages the model–tool interaction loop, while specialized navigation and manipulation policies execute physical skills. Ludi$_{0.1}$ demonstrates a practical path toward fluid human–robot collaboration today while producing the multimodal interaction traces needed to develop a more deeply integrated foundation model for robots and people.

Videos and additional examples: ludorobotics.ai

1. Introduction

Recent advances in robot foundation models have substantially expanded the range of tasks robots can perform. Models built on pretrained large language and vision–language models exhibit increasingly general perception, reasoning, and language-conditioned control across navigation and manipulation (Driess et al., 2023; Zitkovich et al., 2023; Kim et al., 2024b; Black et al., 2024; Bjorck et al., 2025; Cheng et al., 2024), while more recent approaches have begun to incorporate generative world models to support prediction and action generation (Zhen et al., 2024; Ye et al., 2026; Zhang et al., 2025). Much of this progress, however, has emphasized physical task competence and generalization. Natural human–robot collaboration requires capabilities beyond physical competence: robots must also interpret human intent, communicate naturally, and adapt their behavior through speech, gesture, and shared context.

Consider a household task performed by a Unitree G1 humanoid robot, illustrated in Figure 1. A person asks the robot:

> "Hey Ludi, could you bring Chloe a Coke?"

Two soda cans sit on the table, one Coke and one Pepsi. The request names which one, so the robot says "Sure," and checks whether the red can is within arm's reach and reaches out to pick it up.

A challenge arises when the user's intent changes after the robot has already begun acting. Suppose the user initially chooses Coke, and Ludi$_{0.1}$ begins reaching for the Coke. The user then remembers that Chloe prefers Pepsi and says:

> "Oh, wait. She actually prefers Pepsi."

Ludi$_{0.1}$ must understand that this correction supersedes the earlier choice, relate it to the action already underway, and revise its behavior accordingly. It abandons the reach for the Coke, picks up the Pepsi, navigates to the bedroom, and delivers it to Chloe's desk.

This example illustrates a key form of social intelligence that is natural for people but difficult for robots: adapting quickly when a user's intent or instruction changes after action has begun. Doing so requires speech, perception, memory, reasoning, navigation, and manipulation to operate as an integrated system, allowing new information to reshape ongoing behavior. Despite recent progress in robotics, making such fluid, responsive interaction reliable across varied environments and tasks remains an open challenge. This work takes a step toward that goal by integrating social interaction and physical behavior within a single robotic system.

[†]The complete list of authors is in the Authorship section at the end of this report.

STEP	VISUAL INPUT	USER INPUT & TOOL CALL OUTPUT

01	**Pick red can**
	BEDROOM LIVING ROOM EGO
	 
		USER Jake: "Hey Ludi, could you bring Chloe a Coke?"
		say t+6.4s
		`Sure.`
		reachability t+7.5s
		`red Coca-Cola can | right arm`
		vla RUNNING t+9.0s
		`pick red can`
		The robot reaches for the Coca-Cola can.

02	**Pick blue Pepsi can, succeeded**
	BEDROOM LIVING ROOM EGO
	 
		USER Jake: "Oh, wait. She actually prefers Pepsi."
		vla CANCELLED t+20.8s
		`pick red can`
		say t+24.3s
		`Got it. Pepsi instead.`
		reachability t+26.0s
		`blue Pepsi can | right arm`
		vla SUCCEEDED t+28.0s
		`pick blue Pepsi can`
		Jake speaks mid-action, so the pick is cancelled. The robot stops reaching for the Coke and picks up the Pepsi.

03	**Move to bedroom, desk** BEDROOM LIVING ROOM EGO 	**lookup_keypoints** t+44.0s `bedroom | desk` **navigate SUCCEEDED** t+45.5s `bedroom | desk` Pepsi in hand, the robot looks up the desk and moves to the bedroom to find Chloe.
04	**Place Pepsi beside laptop, succeeded** BEDROOM LIVING ROOM EGO 	**vla SUCCEEDED** t+66.0s `place Pepsi beside laptop` The robot sets the Pepsi down on the desk where Chloe is working.
05	**Robot says Jake asked me to bring you this.** BEDROOM LIVING ROOM EGO 	**say** t+90.0s `Jake asked me to bring you this.` The robot explains the delivery to Chloe.

Figure 1 Social intelligence in action. In this real-robot demonstration on a Unitree G1 humanoid, a user asks Ludi$_{0.1}$ to bring a Coke to Chloe in the bedroom, and Ludi$_{0.1}$ begins reaching for the Coke. When the user remembers that Chloe prefers Pepsi and changes the request, the robot abandons the original action, retrieves the Pepsi, navigates to the bedroom, and delivers it to Chloe's desk. The episode illustrates how dialogue, contextual reasoning, navigation, and manipulation are coordinated within a single evolving human–robot interaction.

2. Ludi$_{0.1}$: An Agentic Approach

Ludi$_{0.1}$ integrates robot perception, navigation, and manipulation with interactive speech, dialogue, memory, and social reasoning in a functioning robotic system. Our current implementation runs on a Unitree G1 humanoid. We deliberately chose an agentic architecture rather than relying on a single end-to-end model because natural human–robot interaction requires the robot to coordinate specialized capabilities over time and revise ongoing behavior as new information arrives.

Modern vision–language models (VLMs) jointly represent visual and linguistic information and support rich visual–language understanding and open-ended multimodal generation (Radford et al., 2021; Alayrac et al., 2022; Liu et al., 2023b). These capabilities make VLMs a natural candidate for the reasoning core of an interactive robot. At the center of Ludi$_{0.1}$ is a fine-tuned VLM that serves as the reasoning core of the agent. It interprets the current scene and interaction history and decides what the robot should say or do next, including whether to wait, navigate, or invoke a manipulation policy.

Ludi$_{0.1}$ operates within a harness that manages the interaction loop. The harness assembles the current context, calls the VLM, validates and executes its selected tool, records the result, and calls the VLM again with the updated context. A single user request can therefore produce several cycles of reasoning, tool use, and observation before the interaction returns to the user. The VLM decides what should happen next; the harness manages how that decision is executed and incorporated into the ongoing interaction.

Spoken input triggers a new agent turn, while speech output runs asynchronously so that reasoning and action can continue while the robot is speaking. New user input can arrive at any time and redirect the robot even while it is speaking or acting. The harness maintains a shared interaction record containing dialogue, visual observations, model decisions, tool calls, and outcomes. The VLM reasons over this record and generates new plans, responses, and action decisions. As the interaction grows, an auxiliary model compacts older context in the background without disrupting an active turn. All inference and system operation can run locally, without relying on external model APIs or cloud-based inference. In our experiments, Ludi$_{0.1}$ ran on an onsite GPU workstation connected to a Unitree G1 robot.

We refer to the complete embodied agent as Ludi$_{0.1}$: the VLM together with the harness, speech components, navigation system, and manipulation tools, depicted in Figure 2. For manipulation, Ludi$_{0.1}$ invokes specialized vision–language–action (VLA) policies, a class of models that conditions on visual observations and language instructions to generate robot actions (Zitkovich et al., 2023; Kim et al., 2024b). In our current implementation, these manipulation policies are based on GR00T N1.7 (Bjorck et al., 2025). Within its current set of supported environments and skills, Ludi$_{0.1}$ can engage in spoken interaction, interpret the surrounding scene, ask clarifying questions, navigate indoors, and carry out manipulation tasks using these policies. It can maintain context across an extended interaction and revise an ongoing plan when the user provides new information.

Together, these capabilities allow Ludi$_{0.1}$ to handle interactions that cannot be reduced to isolated perception, dialogue, navigation, or control tasks. The robot can move fluidly between understanding a request, gathering missing information, explaining its intent, acting in the physical world, and revising its behavior in response to the user. The goal is not simply to make robots more conversational. Speech must remain grounded in what the robot sees, remembers, plans, and does, so that communication and physical action unfold as one coherent interaction in the real world.

3. A VLM for Interactive Robotics

At the center of Ludi$_{0.1}$ is a Qwen3.5 vision–language model (VLM) (Qwen Team, 2026), fine-tuned specifically to serve as the reasoning core of the agent by perceiving the scene, interpreting and responding to user interactions, and deciding which tool to invoke at each step. Starting from the pretrained model, we fine-tuned it on synthetic multi-turn demonstrations of complex tasks and human–robot interactions. Our objective was to go beyond visual recognition and instruction following, training the model to interpret an evolving interaction, reason over shared context, select among speech and physical tools, and revise its behavior when the user provides new information.

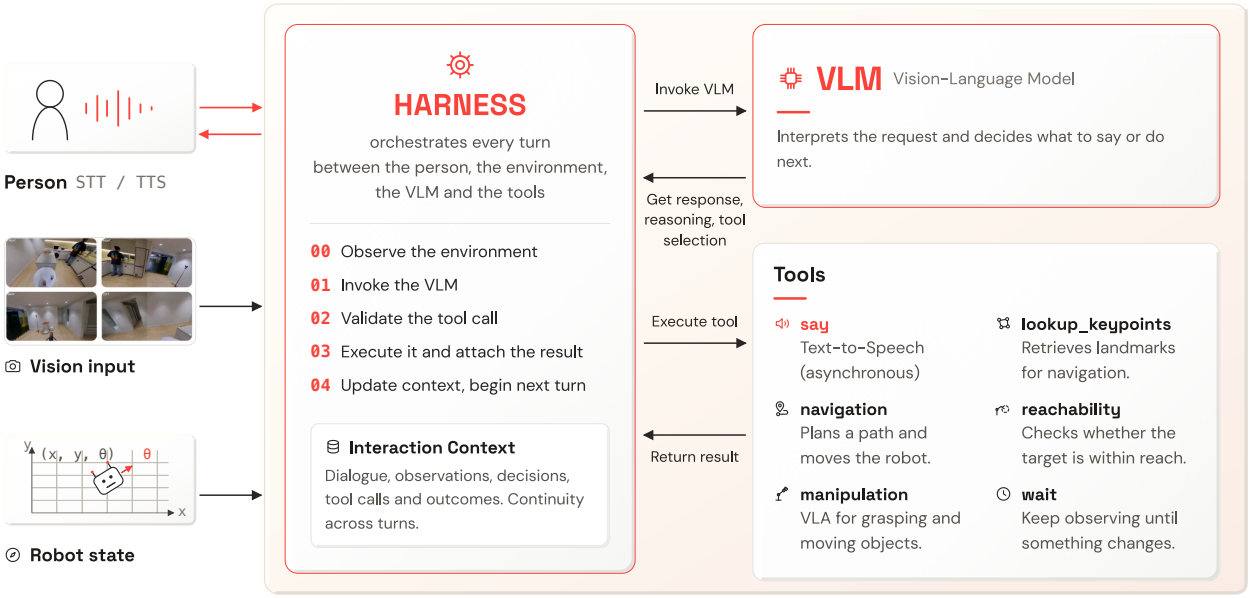

Figure 2 Architecture of the Ludi$_{0.1}$ embodied agent. The VLM provides scene perception, reasoning, dialogue, and tool selection, while the harness manages model calls, tool execution, results, and shared context across turns.

3.1. The Agent Interface

At each turn, the model receives the robot's ego-view image, panoramic context from a head-mounted 360° camera, the transcribed user speech, the interaction history including its own reasoning, and the results returned by previous tool calls. It responds with a single action drawn from a small, fixed tool set:

- `say`: speak to the user, including responding to the user and asking a clarification question.
- `lookup_keypoints`: retrieve the static map of rooms and named locations, sorted by distance from the robot.
- `navigate`: move to a named location in the environment.
- `reachability`: check whether a visible object can be picked up, and which arm to use, before attempting manipulation.
- `vla`: execute a manipulation, including picking up or placing an object via a trained VLA policy.
- `wait`: hold and keep observing until something changes.

Each tool is exposed to the model through a typed function-call schema, a structured interface that specifies the tool's arguments and their allowed values. The harness enforces the contract around each call: a turn may contain at most one action-bearing tool call, a pick must be directly preceded by a successful `reachability` check, and malformed or out-of-contract calls are rejected rather than executed. Parts of these schemas are populated at runtime. The `object_type` vocabulary of `reachability` and `vla` is read from the robot's VLA registry when the agent starts, so newly trained manipulation skills automatically appear to the model as new argument values. This allows the set of manipulation skills available to the agent to expand without retraining the agent itself.

3.2. Training Data and Fine-Tuning

We built the training corpus by generating interaction scenarios combinatorially over tasks, objects, and interaction events. Each scenario was rendered into a multi-turn interaction trajectory and serialized as a tool-call trace with intermediate reasoning. The corpus covers the situations that make interactive robotics hard: ambiguous requests, mid-task corrections, interruptions, multi-step errands, and dialogue that mixes task-oriented conversation with social interaction. The training corpus consisted of 14,432 trajectories.

The demonstrations were fully synthetic, with tool-call correctness guaranteed by construction: a state machine and policy table determined every tool call, its arguments, and its result. A teacher LLM wrote each turn's language, including the reasoning text, spoken responses, and user utterances, controlling vocabulary and style. Every authored turn then passed through a quality gate of automated rule-based checks and a two-stage LLM judge. If a turn failed these checks, the system allowed up to two constrained revision attempts; turns that still failed were rejected. Scene imagery comes from generated images and simulator renders of environments that mirror the deployment space. Figure 3 shows a generated demonstration.

Figure 3 An example of generated data. The user identifies the target by spatial position ("the *middle* laptop"); the model grounds the target in the four-view observation before issuing a reachability check. The multi-turn tool history (CONTEXT) and the latest four-view observation form the input, while the reasoning and one action (IMG + TURN) are the supervision target.

We explored several adaptation strategies, including distillation and prompt optimization, and ultimately used supervised fine-tuning on the interaction traces. We fine-tuned Qwen3.5 at the 4B and 9B scales (Qwen Team, 2026). The fine-tuned 9B model achieved higher benchmark success than the 4B model, but we deployed the smaller model because its lower inference latency provided a better tradeoff for real-time interaction.

3.3. Closed-Loop Simulation for Interactive Robot Agents

Existing VLM benchmarks typically score isolated predictions: given an image and a question, compare the model's answer against a reference answer or predefined choice (Goyal et al., 2017; Hudson and Manning, 2019; Yue et al., 2024). That misses what an embodied agent actually has to do, which is to sustain a multi-turn interaction with people while acting in a world that changes because of its own actions. We therefore built a closed-loop simulation environment in which the agent serves a user end-to-end: it perceives, speaks, navigates, and manipulates through exactly the same tool interface and system prompt as on the real robot. The only scripted component is the simulated user; the agent's behavior and the resulting evolution of the environment unfold in closed loop.

We run these closed-loop interactions in AI2-THOR (Kolve et al., 2017), an interactive indoor simulator, using ProcTHOR (Deitke et al., 2022) to generate diverse 3D homes with different room layouts, furnishings, and object placements. Each generated home is fixed by hash so that every scenario can be reproduced exactly. The simulated robot matches the deployed system: the agent receives the same four panoramic views using the same calibrated camera geometry as the robot's head-mounted 360° camera, navigates only to named keypoints on a static map, and manipulates through the same pick-and-place interface.

3.4. Evaluation

We designed Interaction Core28, a closed-loop benchmark in AI2-THOR consisting of 28 scenarios that cover delivery, perception and reporting, ambiguity that requires clarification, robustness to failed searches and unreachable objects, and mid-task interaction. Each scenario defines a gold tool-call trajectory, and every episode is scored on both its interaction and its final simulator state. Core28 is our primary measure because it exercises the full loop of perception, dialogue, and action.

An episode counts as a task success when it terminates cleanly, satisfies the scenario's deterministic interaction gates, and reaches the required simulator end state. The gates encode the interaction behavior being tested, such as asking for clarification before acting under ambiguity or reporting honestly when a target cannot be reached.

Because deterministic gates cannot reliably account for alternative phrasings, a failed episode is passed to an LLM judge when the sole failing gate concerns the semantic content of a spoken utterance. The judge examines the full transcript, staged ground truth, and scenario success criterion and may overturn the failure to a pass. It cannot overturn failures involving tool use, task execution, or the final simulator state.

Table 1 reports episode-level task success on Core28. Fine-tuning roughly doubles success over the same pretrained model, and the fine-tuned 9B completes 20 of 28 scenarios end-to-end, approaching a frontier model (GPT-5.6) with an optimized prompt. Remaining failures concentrate in spatial grounding at the delivery end of a task: the agent picks the right object and plan but localizes the hand-off point imprecisely.

Table 1 Interaction Core28 closed-loop scenario benchmark (simulation). Episode-level task success after LLM-judge appeal; the parenthesized count is how many of those successes the judge granted by overturning a language-semantic gate failure. Fine-tuned models use LoRA (Hu et al., 2022) applied to the corresponding Qwen3.5 base models.

Model	Success (judge-overturned)
GPT-5.6 (GEPA-optimized prompt; Agrawal et al. (2026))	26/28 (0)
Qwen3.5-4B (vanilla)	10/28 (1)
Qwen3.5-4B (fine-tuned)	**17/28** (1)
Qwen3.5-9B (vanilla)	11/28 (0)
Qwen3.5-9B (fine-tuned)	**20/28** (1)

4. Engineering the Harness for Real-Time Interaction

Ludi$_{0.1}$'s control harness is designed to keep the interaction responsive and reliable even when model inference, context management, speech, and physical actions operate at different timescales. A few design choices are particularly important:

- **Non-blocking context compaction:** As the interaction history grows, an auxiliary model summarizes older context in the background. The compacted history is installed only between active turns, so context management never interrupts or changes the state of a running agent loop.
- **Visual prefill and asynchronous speech:** A background process continually refreshes the camera input and performs visual prefill before a user instruction arrives, reducing the delay before the VLM can respond. Spoken responses are placed on a FIFO audio queue, allowing the agent loop to continue while ensuring that utterances play completely and in the order they were generated.
- **Responsive Local Speech Interaction:** Ludi$_{0.1}$ uses a locally deployed speech pipeline optimized for fast human–robot interaction. In our benchmark, once a recorded 10-second English utterance had been captured, the selected four-thread sherpa-onnx configuration (The k2-fsa Team, 2023) transcribed it in 94.7 ms on average, with 99.2 ms P95 processing latency and no transcription errors on the test clips. These measurements reflect STT processing time rather than end-to-end human-perceived response latency. More important for the overall system is how speech is integrated into the agent loop: transcripts automatically start new agent turns, speech output runs asynchronously without blocking reasoning or action, queued playback prevents utterances from interrupting one another, and new user input can redirect ongoing navigation or manipulation.

5. Humanoid Navigation: Keypoints and Continuous Planning

Ludi$_{0.1}$ uses a navigation stack that combines conventional localization and keypoint-based navigation with a continuous planner for motions requiring finer humanoid-specific control. Localization combines KISS-ICP LiDAR odometry (Vizzo et al., 2023) with an extended Kalman filter, with hyperparameters tuned automatically using Optuna (Akiba et al., 2019). For most household tasks, the VLM does not reason directly over metric coordinates. Instead, it navigates through a static map of named keypoints associated with semantically meaningful locations such as rooms, desks, tables, and work areas. The agent can query the available keypoints and then issue a navigation command using the location name. This provides a simple interface between high-level language reasoning and the lower-level navigation stack while keeping metric planning outside the VLM. Before invoking a manipulation skill, the agent uses the `reachability` tool to determine whether the target object is within arm's reach. This creates a natural handoff between navigation and manipulation: navigation brings the robot to a semantically meaningful location, while the reachability check determines whether the object can be acted on from the current pose.

When finer local motion is needed, the system invokes an orientation-aware continuous planner tailored to the G1 humanoid. The planner represents position and orientation with quintic spline trajectories, with heading planned independently of direction of travel. It uses an orientation-aware multi-circle footprint to account for the robot's body geometry during turning, strafing, backward motion, and passage through narrow spaces. The same trajectory representation supports planning, control, replanning, and safety checking, allowing the robot to replan smoothly without repeatedly stopping and restarting.

6. Specialized VLA Manipulation Policies

Physical manipulation in Ludi$_{0.1}$ is carried out by specialized VLA policies invoked as tools by the agent. For the Ludi$_{0.1}$ demonstration, we fine-tuned GR00T N1.7 (Bjorck et al., 2025) on the manipulation skills needed in the target home environment, as illustrated in Figure 4. The VLA setup has four key features:

- **Pretrained G1 embodiment:** We initialized from GR00T N1.7's pretrained Unitree G1 embodiment components (`UNITREE_G1`), including the embodiment-specific state and action representations, and fine-tuned the model directly on our task-specific demonstrations.
- **Arm-and-gripper control:** The VLA controls the robot's manipulation degrees of freedom only, predicting arm and gripper actions rather than whole-body motion. Navigation and other body motion are handled separately by the robot's locomotion stack.
- **Chunked action prediction:** The VLA predicts and executes an action horizon of 30 steps synchronously at each inference call. With the robot operating at 30 Hz, each chunk corresponds to approximately one second of motor commands before the next VLA inference.
- **Real-robot fine-tuning:** Fine-tuning data were collected through teleoperation on the physical G1 in household pick-and-place settings similar to those used in the demonstration. This grounds the pretrained model in the robot's actual visual observations, kinematics, and manipulation behavior.

These VLA policies serve as learned manipulation skills rather than as the system's primary reasoning component. The VLM interprets the user's request and the broader interaction, decides when manipulation is appropriate, and invokes a VLA with a task instruction. The VLA executes the physical behavior and returns its outcome to the agent when the episode completes or is interrupted.

Calibrated simulation environment. Although the final VLA policies were fine-tuned only on real-robot demonstrations, we developed a high-fidelity Isaac Sim model of the G1 (NVIDIA Isaac Sim Project Developers, 2026) to accelerate policy development and evaluation before physical-robot rollouts. Simulation provided a repeatable testbed for comparing model architectures, training procedures, execution horizons, action representations, and data mixtures, while closed-loop rollouts exposed policies that had low training loss or strong open-loop metrics but failed to complete the task.

To make these comparisons meaningful, we calibrated system timing, camera pose, scene geometry, and motor dynamics separately against recorded behavior of the physical robot. Camera calibration achieved a

Figure 4 Example of a real-robot pick-and-place task using the learned VLA policy. The Unitree G1, equipped with a Dex-3 hand, grasps the target can from the tabletop using egocentric visual observations.

mask intersection-over-union (IoU) of approximately 0.92, while Dex-3 hand tracking error decreased from 0.132 to 0.043 rad (∼7.6° to 2.5°) with fingertip error reduced to at most 13 mm. The fitted dynamics also generalized to 17 held-out episodes from other tasks. We therefore used simulation as a screening tool rather than a replacement for real-robot evaluation: because simulated performance did not always preserve policy rankings on the physical robot, promising policies were ultimately validated through real-world rollouts.

7. Related Work

A growing line of work uses large language or vision–language models as high-level decision makers over grounded robot skills. SayCan combines a language model's estimate of an appropriate next action with learned affordance values that reflect what the robot can actually execute (Ahn et al., 2022). Inner Monologue closes this loop by returning scene descriptions, success signals, and human feedback to the model so that it can revise its plan (Huang et al., 2022). Code as Policies instead uses a code-generating language model to compose perception and control APIs directly. Together, these systems illustrate an early agentic pattern in robotics: a general model selects and composes actions, while specialized components execute them and report the results (Liang et al., 2022). More recent systems make this decomposition more explicit. Agentic Robot (Yang et al., 2025) coordinates a reasoning model, VLA executor, and temporal verifier in a closed planning–execution–verification loop, while VLAs-as-Tools (Lei et al., 2026) uses a high-level VLM agent for scene analysis, planning, and recovery, with specialized VLA policies carrying out individual physical subtasks.

A closely related agentic paradigm has been explored with virtual embodiment in PUBG Ally, where an autonomous agent reasons over player input and game state, selects among tools for speech, perception, memory, and action, and iteratively decides what to do next (KRAFTON AI, 2026). PUBG Ally provides a close precedent for the architecture considered here, but with embodiment in a virtual rather than physical environment.

Other work has focused more directly on the interactive challenges that arise when robots work with people. KnowNo calibrates the uncertainty of a language-model planner so that the robot can ask for help when a request is ambiguous (Ren et al., 2023). HELPER-X uses a memory-augmented language model across several interactive embodied domains, including dialogue-based task execution, instruction following, and active question asking (Sarch et al., 2024). Related work has also explored language as a mechanism for intervening in ongoing robot behavior: RT-H (Belkhale et al., 2024) supports language-based corrections during policy execution, OLAF (Liu et al., 2023a) uses verbal corrections to refine robot behavior after deployment, and

Hi Robot (Shi et al., 2025) uses a hierarchical VLM/VLA architecture to incorporate complex instructions and situated user feedback as a task unfolds. These efforts are closely related to Ludi$_{0.1}$'s emphasis on clarification, evolving user intent, and memory across turns. More broadly, recent HRI work shows that LLM-powered robots create interaction requirements distinct from text- and voice-only agents, including stronger expectations for nonverbal behavior (Kim et al., 2024a).

In parallel, VLA foundation models such as RT-2 and GR00T learn robot policies conditioned on visual observations and language instructions (Zitkovich et al., 2023; Bjorck et al., 2025). A particularly close contemporary parallel is Gemini Robotics ER 2, an embodied-reasoning model that can communicate with users, plan and monitor multi-step tasks, and invoke lower-level VLA or navigation capabilities as tools (Hansen and Xu, 2026).

Ludi$_{0.1}$ lies at the intersection of these directions but emphasizes training for sustained human–robot interaction. Where several early agentic systems relied primarily on prompting general models, we fine-tuned the VLM on multi-turn traces containing ambiguity, clarification, social dialogue, interruptions, corrections, and tool outcomes. The resulting VLM reasons over transcribed speech, multi-view visual input, interaction history, and prior action outcomes, while a purpose-built harness manages low-latency execution across speech, navigation, and GR00T-based manipulation policies. Rather than treating agentic systems and robot foundation models as competing alternatives, we are advancing a practical agentic system today while using its interaction traces to train a more deeply integrated foundation model for robots and people.

8. The Limits (and Promise) of Agentic Robotics

Ludi$_{0.1}$ is modular at the execution layer but integrated at the interaction layer. A central VLM reasons over the shared interaction context and decides whether the robot should speak, clarify, wait, navigate, or manipulate, while the harness manages the interaction loop and specialized components execute the selected actions. This architecture enables real-time clarification, handling of interruptions, memory-informed behavior, and coordination between conversation and physical action.

But this integration is operational rather than fully learned. Speech, memory, navigation, and manipulation remain distributed across separate components, and coordinating them through model calls and tools can introduce context loss, latency, brittle handoffs, and fragmented behavior. Although the agent reasons over a shared interaction history, the system does not learn a unified representation of the user, the environment, and the robot's evolving physical state.

We are therefore pursuing two complementary paths. First, we will continue advancing the agentic architecture, improving the VLM, harness, memory, tools, and interaction loop to make Ludi more capable, responsive, and reliable today. Second, we will use the multimodal interaction traces produced by Ludi$_{0.1}$, connecting perception, speech, human feedback, decisions, and physical action, to develop a more deeply integrated foundation model for robots and people.

9. Toward Ludi 1.0: A Foundation Model for Robots and People

The next step in this effort is Ludi 1.0, a foundation model for robots and people that aims to more deeply integrate perception, dialogue, memory, reasoning, and control. Rather than treating conversation as an interface layered on top of robot behavior, Ludi 1.0 aims to ground language and physical action in a shared representation, allowing each to continually inform the other.

Such a model must jointly represent the evolving physical state of the robot and environment, the history and intent of the user, and the relationship between language and action. It should understand how these quantities evolve together over time: what has already happened, what the robot is currently trying to accomplish, what the user expects, and how new observations or utterances change the appropriate course of action. This representation should support not only choosing what to do next, but recognizing when an action is no longer appropriate, when additional information is needed, and when to explain, clarify, revise, pause, or stop as the interaction unfolds.

Ludi$_{0.1}$ is therefore both a working embodied agent and a testbed for developing these capabilities. By

exposing the limits of current architectures and collecting traces of human–robot interaction, it helps reveal what a future foundation model must learn to represent. Ultimately, the goal is a robot whose physical and social intelligence are deeply integrated: one that understands people, communicates naturally, and acts as a collaborative partner in one coherent, ongoing interaction.

10. Authorship

Ludi$_{0.1}$ was developed by the following contributors at Ludo Robotics:

Wooseong Chung, William Cong[*,2], Jakub Dworakowski, Ethan Ewer[*,4], Tri Wahyu Guntara[4], Yeonwoo Jeong, Tianchong Jiang[*,3], Chaewon Kim[4], Hyunseo Kim[*], Jinwoo Kim[4], Jinyeon Kim[4], Yea-Seul Kim, Jack Kunde[*], Kangwook Lee[4], Sangheon Lee[*], Robert Nowak, and Junha Roh.

[*]Intern. [2]Also University of Wisconsin–Madison. [3]Also Toyota Technological Institute at Chicago. [4]Also KRAFTON.

References

L. A. Agrawal, S. Tan, D. Soylu, N. Ziems, R. Khare, K. Opsahl-Ong, A. Singhvi, H. Shandilya, M. J. Ryan, M. Jiang, et al. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pages 8479–8565, 2026.

M. Ahn, A. Brohan, N. Brown, et al. Do as I can, not as I say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, 2019. doi: 10.1145/3292500.3330701.

J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:23716–23736, 2022.

S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh. RT-H: Action hierarchies using language. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.049.

J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

A.-C. Cheng, Y. Ji, Z. Yang, Z. Gongye, X. Zou, J. Kautz, E. Bıyık, H. Yin, S. Liu, and X. Wang. NaVILA: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024.

M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi. ProcTHOR: Large-scale embodied AI using procedural generation. *Advances in neural information processing systems*, 35:5982–5994, 2022.

D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence. PaLM-E: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6325–6334, 2017.

S. Hansen and P. Xu. Introducing Gemini Robotics ER 2, July 2026. Google DeepMind.

E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

W. Huang, F. Xia, T. Xiao, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

D. A. Hudson and C. D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6700–6709, June 2019.

C. Y. Kim, C. P. Lee, and B. Mutlu. Understanding large-language model (llm)-powered human-robot interaction. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pages 371–380, 2024a. doi: 10.1145/3610977.3634966.

M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 2024b.

E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, et al. AI2-THOR: An interactive 3d environment for visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

KRAFTON AI. From workflow-based slm to autonomous agent: Evolving pubg ally's architecture. KRAFTON AI Blog, Apr. 2026. URL https://www.krafton.ai/blog/posts/2026-04-15-pubg_ally_nemotron/pubg-ally-nemotron-en.html.

Z. Lei, C. Liu, Y. Xiong, M. Xiong, Y. Ding, Z. Zhang, W. Li, and S. Chen. Towards long-horizon embodied agents with tool-aligned vision-language-action models. *arXiv preprint arXiv:2605.13119*, 2026.

J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

H. Liu, A. Chen, Y. Zhu, A. Swaminathan, A. Kolobov, and C.-A. Cheng. Interactive robot learning from verbal correction. *arXiv preprint arXiv:2310.17555*, 2023a.

H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023b.

NVIDIA Isaac Sim Project Developers. NVIDIA Isaac Sim: An Open-Source Reference Framework for Robotics Simulation, Testing, and Synthetic Data Generation, 2026. URL https://github.com/isaac-sim/IsaacSim. Built on NVIDIA Omniverse.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

A. Z. Ren, A. Dixit, A. Bodrova, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

G. Sarch, S. Somani, R. Kapoor, M. J. Tarr, and K. Fragkiadaki. HELPER-X: A unified instructable embodied agent to tackle four interactive vision-language domains with memory-augmented language models. *arXiv preprint arXiv:2404.19065*, 2024.

L. X. Shi, B. Ichter, M. R. Equi, L. Ke, K. Pertsch, Q. Vuong, J. Tanner, A. Walling, H. Wang, N. Fusai, A. Li-Bell, D. Driess, L. Groom, S. Levine, and C. Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 54919–54933, 2025.

The k2-fsa Team. Sherpa-onnx: Speech-to-text, text-to-speech, and speaker verification using next-gen kaldi and onnx runtime. https://github.com/k2-fsa/sherpa-onnx, 2023.

I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss. KISS-ICP: In defense of point-to-point ICP – simple, accurate, and robust registration if done the right way. *IEEE Robotics and Automation Letters*, 8(2):1029–1036, 2023. doi: 10.1109/LRA.2023.3236571.

Z. Yang, Y. Chen, X. Zhou, J. Yan, D. Song, Y. Liu, Y. Li, Y. Zhang, P. Zhou, H. Chen, et al. Agentic Robot: A brain-inspired framework for vision-language-action models in embodied agents. *arXiv preprint arXiv:2505.23450*, 2025.

S. Ye, Y. Ge, K. Zheng, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, Z. Ge, S. Stevens, D. Jiang, W. Ren, Y. Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

W. Zhang, H. Liu, Z. Qi, Y. Wang, X. Yu, J. Zhang, R. Dong, J. He, H. Wang, Z. Zhang, et al. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, 38:24195–24228, 2025.

H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan. 3D-VLA: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

B. Zitkovich, T. Yu, S. Xu, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 2023.