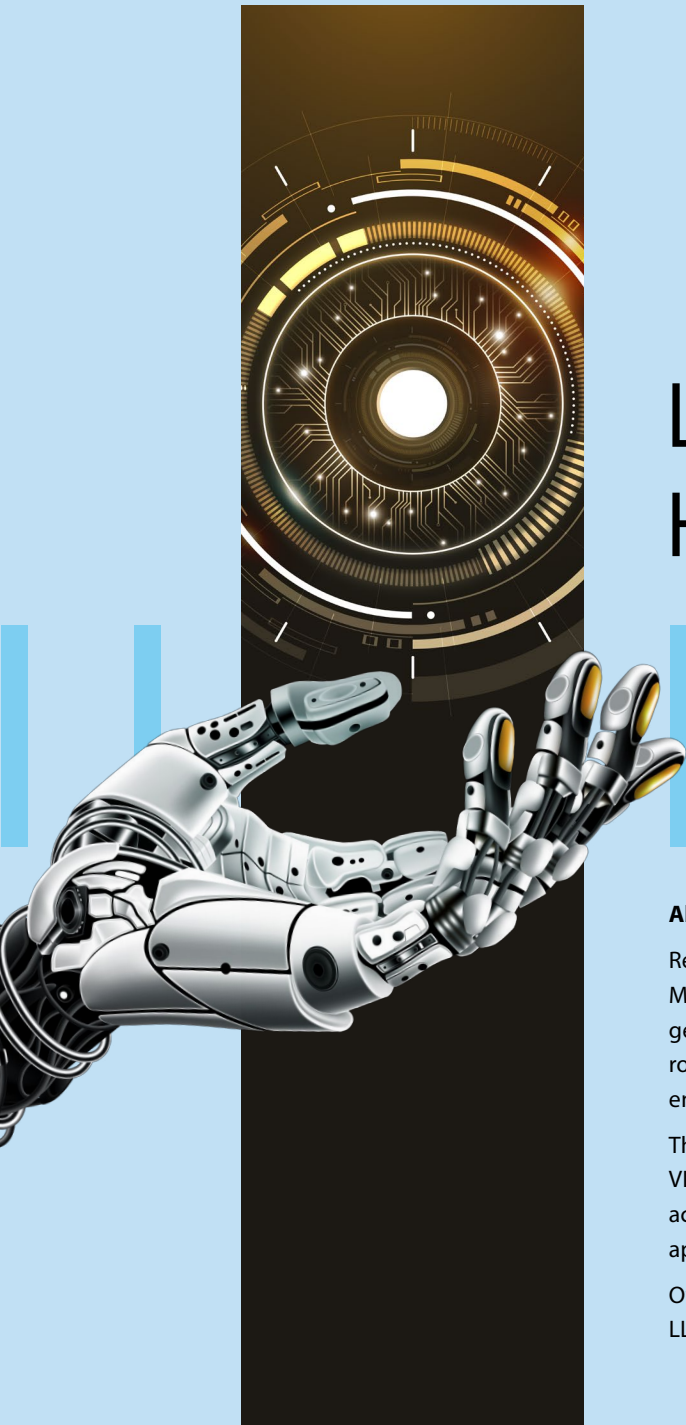




LLM'S AS COGNITIVE ENGINES FOR HUMANOIDS

Abstract

Recent advances in Large Language Models (LLM's) and Vision-Language-Action Models (VLA's) are transforming robotics from task-specific systems into more general-purpose humanoids. These models act as a cognitive layer, enabling robots to understand human intent, reason over tasks, and operate in real-world environments.

This work explores the evolution of robotic policies, the architecture of modern VLA-based systems, and the broader cognitive roles played by LLM's beyond action generation. It also highlights key challenges of using LLM's with alternative approaches like world models and robot-only training.

Overall, this perspective outlines both the potential and current limitations of using LLM's as cognitive engines for humanoid robots.

Keywords

Large Language Models (LLM's), Vision-Language-Action Models (VLA's), Humanoid Robots, Embodied AI, Robotics Policies, Cognitive Architectures, Generalist Policies, World Models

1. Introduction

1.1. Rise of Robotics Industry

Robotics is a well-established domain, having been around for over sixty years. However, we are currently experiencing an extraordinary expansion, as companies develop robots to automate intricate industries and engage in customer-oriented services. The key factor driving this transformation is the advancement of Large Language Models (LLMs). LLMs are revolutionizing the robotics sector by enabling effective understanding, perception, and communication across various modalities. They can analyze visual and auditory data, reason based on that information, and pursue specific objectives. LLMs have started to function as a cognitive layer, breaking down user intentions into subgoals that inform robot control strategies. Recently, Vision-Language-Action (VLA) models have advanced this concept by directly managing robot actions based on visual and textual inputs. The discipline has progressed from tailored, single-task policies, each crafted for a particular robot and action and unable to generalize across different tasks or forms, to integrated models. Where previously multiple tasks necessitated several policies interconnected through routing logic, VLAs now act as versatile systems capable of directly managing a diverse array of robot forms, whether it be a Spot, a humanoid, or a singular robotic arm. [1][6][7]

1.2. Understanding LLMs and Their Strengths

LLM's are large neural networks with billions to trillions of parameters, trained on terabytes of text/images from the internet. But it's not just the size that matters. When you scale these models enough, they start showing abilities that nobody explicitly programmed into them, things like reasoning step by step, learning from a few examples you put in the prompt, and following complex instructions. This is what makes them interesting for robotics. You can give an LLM's some tasks it has never seen before, and it can still break it down into a reasonable plan. Now when you extend these models to also take in images and output actions, the same reasoning abilities carry over to the physical world. They can look at a scene, figure out what objects are there, understand how to interact with them, and come up with a plan. That mix of having a lot of world knowledge, being able to reason flexibly, and being extendable to vision and actions is what makes LLM's a strong fit as cognitive engines for robots [2][3].

2. History of robotics policies from rule-based to general-purpose policies:

2.1. Pre LLM's Era:

Before LLM's came into the picture, it was all about rule-based policies created for a static environment to handle a single job. A warehouse robot recognized exactly the SKU's it was trained on, grasped them with hard-coded trajectories, and placed them at fixed viewpoints. If you ask it to do it on a new object, it will stumble and it won't handle the job.

Changing tasks require re-engineering entire pipelines, right from getting the data to creating simulated environments to testing on them, required the entire new pipelines because of the static system it was in.

Those robots were called humanoids, but they were just cosmetically humanoids but internally they were just bots designed to act on a particular workflow.

2.2. What Qualifies a Robot as a Humanoid?

A robot can meaningfully be considered a humanoid when it demonstrates the following characteristics:

- A physical form that functionally resembles a human.
- The ability to understand a user's intent and natural vocabulary, rather than responding only to hard-coded instructions.
- The capacity to reason during task execution, not just before it.
- The ability to compose actions dynamically in response to the environment, rather than replaying pre-scripted responses.
- The ability to operate in environments built for humans, such as homes or service settings like healthcare, and generalize effectively within those contexts.

When a robot can operate competently in spaces where humans perform everyday tasks, and do so with genuine generalization, it begins to warrant the label of humanoid in a meaningful sense.

2.3. Evolution of LLM's in humanoids:

The path of integrating large language models (LLMs) into robotics can be outlined through several significant systems:

SayCan [1]: It was the first time LLM'S (palm: 540B parameters [2]) was used for robots. At this point, the LLM'S was not grounded in the physical environment and functioned solely as a high-level task planner. Upon receiving user instructions, it broke down the task into individual steps that could be executed by existing robot policies. As time progressed, environmental feedback, including indicators of success and failure as well as scene descriptions, was integrated into the prompt to enhance robustness in cases where a step failed. Concurrently, initiatives like Code as Policies adopted an alternative strategy: instead of producing natural language plans, LLM's were prompted to generate executable Python code that directly invoked robot APIs, such as pick, place, and move. This approach provided more accurate control without necessitating new policy training, although the LLM's still did not possess true physical comprehension.

Vision Language Action (VLA)

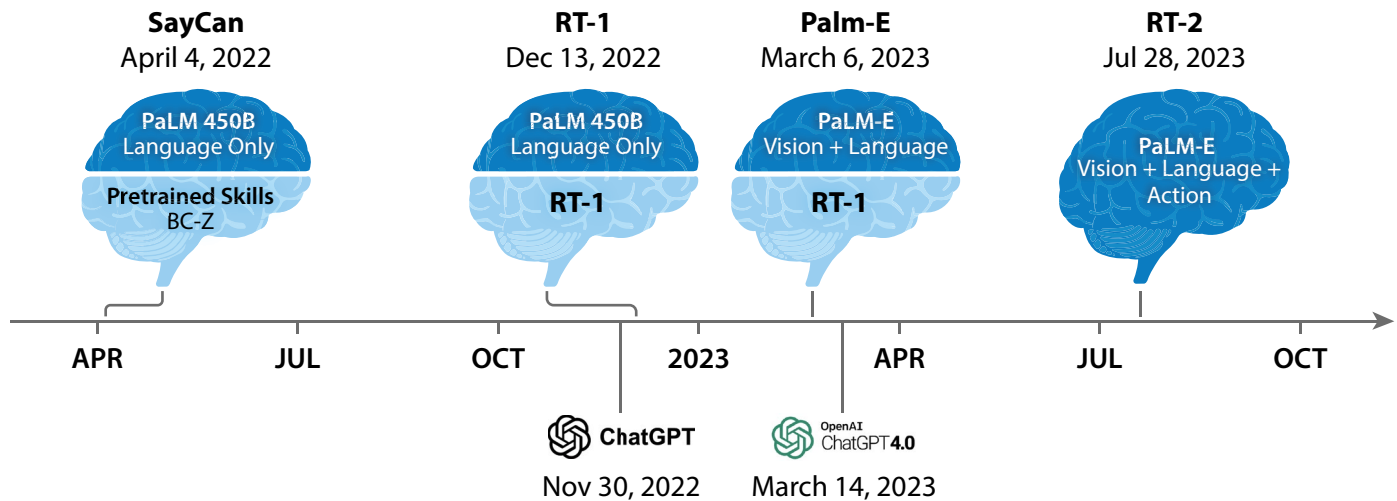


Figure 1: Evolution of Vision-Language-Action (VLA) systems from SayCan to RT-2. Adapted from a YouTube lecture on VLA-based robotics systems by Welch Labs.

RT1(robotics transformer 1) [4]: it was a 35 million parameter transformer, particularly trained on 130K real robot demonstrations of 700+ tasks on a fleet of everyday robots. The RT1 took images (camera images) and natural language instructions as input and predicted robotic actions. It was working well with short or templated commands but could not plan for reason or handle open-ended languages or conversations.

Palm-e (Vision language model) [5]: the 602 billion parameter model that was trained to accept both image and text tokens as an input at scale. With the integration of vision, the model could generate meaningful instructions for robot policies or creating code for motor control. While it continued to function as a high-level planner, it clearly illustrated the critical importance of visual grounding for robots operating in real-world settings.

RT2 [6] marked a pivotal turning point. Utilizing PaLM-E as its vision-language foundation, RT-2 underwent fine-tuning to produce robotic actions directly from both visual and textual inputs. This was the inception of VLAs: a model capable of seeing, reasoning, and acting within a cohesive architecture. The outcomes were significant enough to inspire new research avenues and

enterprises, including Physical Intelligence (π_0). RT-2 illustrated that the extensive world knowledge embedded in internet-scale LLM's pretraining could be applied to physical manipulation, effectively linking language intelligence with embodied intelligence. [6]

RT-2 illustrated that the extensive world knowledge embedded in internet-scale LLM pretraining could be applied to physical manipulation, effectively linking language intelligence with embodied intelligence. [6]

Later, many companies started implementing VLAs and open-source projects like OpenVLA [7], π_0 (Pi-zero) [8], and Octo [9]. They began to be used in humanoids, spots, and all other forms of embodied intelligence. There were many Modifications in the architecture over time. Pi zero created a separate action head which uses flow matching to generate the actions, which are very good and very smooth because there can be many trajectories to reach a particular destination or a position [8].

These same VLA architectures were also utilized for whole-body coordination in humanoid platforms such as the Unitree G1, where a single model managed full-body movement with remarkable stability while responding to natural language commands.

3. End-to-End architecture: from pixels to torques

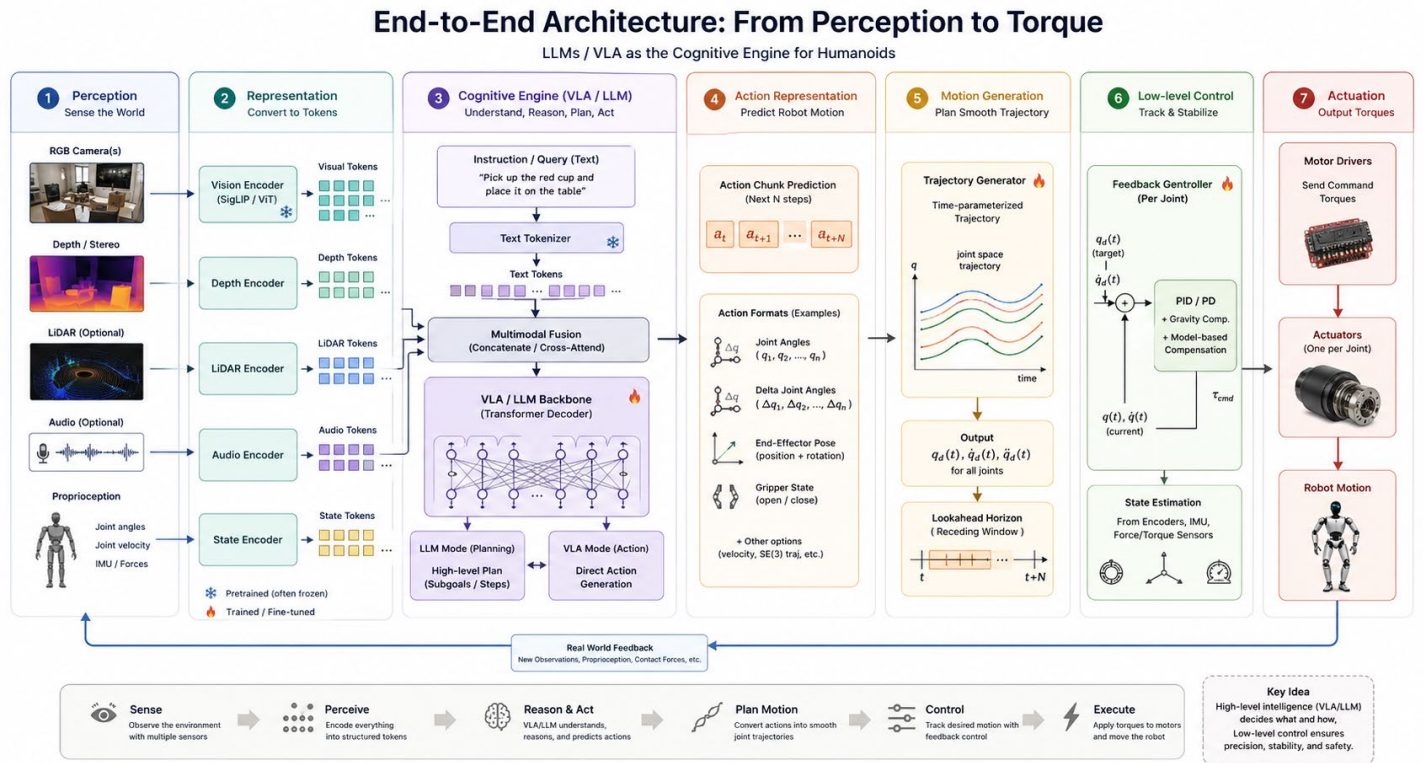


Figure 2: Proposed end-to-end architecture illustrating how multimodal sensory inputs are processed through VLA/LLM cognitive layers to generate robot actions and low-level motor torques.

3.1. Perception:

Multiple sensors or streams feed the system simultaneously.

- RGB cameras
- Depth or light sensors
- Position or joint encoders
- Audio and speech

Visual data is generally resized to dimensions of 224×224 or 336×336 pixels and transformed into visual tokens by a vision encoder such as SigLIP [10], DINOv2 [11], or a specifically fine-tuned variant for tasks. These visual tokens are subsequently combined with language instruction tokens and, if necessary, proprioceptive state information to create the complete input context, which is then forwarded to the VLA to produce motor actions.

The processing pipeline varies based on whether a VLA or a standalone LLM is utilized. A VLA processes this integrated context to generate actions directly. In contrast, an LLM is typically used for high-level planning: it takes the user's instructions and generates a structured sequence of subtasks or steps, which are then sent to lower-level control policies for execution. This LLM-as-planner methodology remains pertinent in scenarios where traditional controllers are highly effective but lack the capability to interpret natural language directly. [1][6]

3.2. Action Generation and Low-Level Control:

VLAs do not actually output the motor torques directly. They predict the target angles for each of N joints for 16-50 sequences all at once using flow matching [8]

Converting these joint angle targets into actual torques is handled by a separate classical control stack, which applies inverse kinematics, trajectory planning, and feedback control. This stack operates at a significantly higher frequency than the VLA, often in the range of hundreds to thousands of Hz and is responsible for translating the VLA's motion intent into stable, precise physical movement. If the limb deviates from the intended trajectory, the controller corrects it in real time while also enforcing constraints such as joint range limits, maximum velocity, and collision avoidance.

This architectural separation exists because neural policies are generally not considered reliable enough for direct torque control. The VLA functions as a low-level semantic planner, deciding what the next action or motion goal should be, while the classical control stack determines how to execute that intent with the precision and stability required in physical environments. Contemporary humanoid systems therefore combine high-level AI reasoning with the rigor of classical control, each operating at the layer it is best suited for. [8][9]

4. Cognitive Role played by LLM's/VLA's:

LLM's and VLA's in humanoid robots serve a far broader function than action generation alone. Motor output is only the most visible surface of their contribution. Their deeper value lies in acting as a cognitive layer situated between perception and control, mediating how the system interprets the world, understands intent, plans behavior, and recovers from failure. [12]

4.1. Perception interpretation:

Vision-language models and multimodal LLM's do not merely identify objects present in a scene. They bring semantic understanding to spatial relationships, object properties, and physical context. A model can determine whether an object is positioned at the edge of a surface or centrally placed, infer whether it is fragile or robust, and adjust its interaction strategy accordingly. This layer of contextual interpretation is what allows robots to move beyond object detection toward genuine understanding. [5] [6]

4.2. Intent and Task Understanding:

Human instructions are rarely precise or unambiguous. LLMs can resolve coreferences, such as identifying the intended referent when a user says, "pick that up," and inferring implicit goals from indirect expressions. A statement such as "I am tired" can prompt the system to consider appropriate responses, whether fetching water, adjusting environmental conditions, or asking a clarifying question. Critically, LLMs can also engage in dialogue to resolve ambiguity before committing an action, reducing the risk of misinterpretation in high-stakes interactions. [1][12]

4.3. Task Planning and Decomposition:

Unlike VLA's, LLM's do not generate motor actions directly. Their primary utility in this context is decomposing a high-level instruction into a structured sequence of subtasks that lower-level control policies can execute. This approach remains particularly relevant in domains where classical controllers are highly capable but cannot interpret natural language. LLM's bridge that gap by translating user intent into a form that precise, physics-aware controllers can act upon. [1][13]

4.4. Memory and Personalization:

The effectiveness of an LLM's is directly tied to the quality of context provided to it. By maintaining records of a user's past interactions, preferences, and the robot's prior errors, the system can retrieve and inject relevant history into the model's context at runtime. This mechanism enables a degree of personalization that is fundamental to making robots feel genuinely useful in human environments. The system should know, for instance, that a user prefers coffee, so that when asked to prepare a drink it can make an informed decision rather than seeking clarification. Similarly, familiarity with a user's environment allows the robot to act on requests efficiently, without needing to re-survey its surroundings each time.

4.5. Error recovery and common-sense reasoning:

The reasoning capabilities of LLM's allow robots to recover from failed states by identifying alternative approaches and continuing toward the original goal. These models also apply commonsense constraints that prevent obviously inappropriate actions. A robot instructed to "drink water" will not attempt to do so, as the model understands this is not a physically meaningful action for a robot. Similarly, safety alignment built into LLM's during training acts as a guardrail against instructions that would involve harmful or unlawful behavior. [12]

4.6. Continual Learning:

Interaction data generated during robot operation can be stored in structured formats and subsequently used for supervised fine-tuning or reinforcement learning on VLA's and LLM's. This allows the system to improve over time, correcting past errors and acquiring new capabilities in a manner loosely analogous to human experiential learning. While the process is less fluid and requires accumulation of sufficient high-quality data before meaningful generalization occurs, it represents a viable pathway toward systems that improve through deployment rather than remaining static after training. [14]

5. Open Challenges in LLM-Based Robotic Systems

Despite their effectiveness across a broad range of tasks, LLM's and VLA's introduce a distinct set of challenges when deployed in physical robotic systems. Their limited grounding in physical dynamics and tendency to hallucinate outside their training distribution remain significant concerns that the field has not yet fully resolved.

5.1. Physical Grounding and Hallucination:

LLM's can propose actions that are physically infeasible or unsafe, particularly when operating at the boundaries of their training data. They are also vulnerable to adversarial inputs such as prompt injection attacks, which can manipulate system behavior in unintended ways. In a robotic context, a single erroneous action can cause irreversible hardware damage, making this a critical failure mode rather than a recoverable one.

World models offer a promising mitigation strategy. By predicting the next environmental state given the current state and a proposed action, world models can serve as a physical prior that filters out unsafe or implausible action proposals before execution. Their stronger grounding in physical dynamics makes them a natural complement to LLMs in safety-critical pipelines. [15]

5.2. Real-time latency:

Running LLM's on local or consumer-grade hardware remains largely infeasible. Even compact VLA's, which are significantly smaller in scale, typically require high-performance GPUs for inference. In practice, latency is partially masked by the fact that VLA's predict 20 to 50 action steps simultaneously. While the current action sequence is executing, the next is being computed in parallel, reducing perceived lag at the motion level. [8]

Latency at the conversational level, however, remains a more persistent issue. Natural human turn-taking in dialogue occurs on the order of 200 milliseconds, a threshold that current LLM-based interaction pipelines rarely meet. Delays compound across multiple stages, including model inference, text-to-speech synthesis, and response parsing, resulting in interactions that feel noticeably less fluid than human conversation.

5.3. Data inefficiency & sim-to-real gap:

LLM's and VLA's are trained predominantly on internet-scale or offline datasets, with relatively limited exposure to real robot interaction data. As a result, they often lack precise understanding of contact dynamics, friction, and the physical edge cases that arise in real-world deployment. A policy that performs reliably in simulation may fail on a physical robot due to subtle mismatches in lighting, surface properties, or object geometry.

Collecting high-quality real-world robot data is costly, time-intensive, and carries inherent risks including hardware wear and safety incidents. Consequently, even when a model has a conceptually correct understanding of a task, execution may fail because it has not been sufficiently exposed to the variation present in real environments. [4][9][14]

5.4. Long-horizon reasoning & consistency issues:

LLM's and VLA's frequently struggle to maintain consistency across tasks that involve many sequential steps. Over extended horizons, the system may lose track of earlier context, alter its plan mid-execution, or generate actions that are locally reasonable but globally inconsistent with prior steps. These small inconsistencies tend to compound, often resulting in task failure even when individual steps appear correct in isolation.

Tasks common to humanoid deployment scenarios, such as cooking, cleaning, or multi-stage assembly, require sustained maintenance of state and intent across long interaction windows. Without robust memory and planning mechanisms, systems are prone to drifting from the original goal, repeating redundant actions, or omitting critical steps entirely. [12][13]

6. Alternatives to LLM's in robotics:

Several viable approaches to building robot intelligence exist that do not rely heavily on LLM's or VLA's, each with distinct architectural philosophies and trade-offs.

One prominent direction is the development of world models, in which the system learns to predict future environmental states from current observations and proposed actions. By capturing the underlying physical structure and dynamics of the world, these models enable more grounded planning and safer decision-making than language-centric systems typically afford. [15]

A separate line of work pursues robot intelligence through reinforcement learning alone, without any dependence on LLM's or internet-scale pretraining. Companies such as Ineffable Intelligence, founded by David Silver, a researcher known for his work on AlphaGo, are exploring this direction by training robotic policies entirely through simulated interaction. The underlying premise is that sufficiently rich simulation environments, combined with RL, can produce capable and generalizable policies without inheriting the grounding limitations or hallucination tendencies associated with language models.

A third approach centers on training policies directly from large-scale robot interaction data, bypassing internet-scale language pretraining entirely. RT-1 [4] demonstrated that paired perception-to-action mappings, learned from real robot demonstrations with

associated language instructions, can yield strong task performance. Generalist AI company extended this direction further, assembling a diverse dataset of over 500,000 robot interaction episodes to train generalist policies from the ground up. Because learning is grounded directly in physical interaction rather than linguistic knowledge, these systems avoid hallucination and grounding errors that commonly affect LLM-based approaches. However, they carry their own limitations: data collection at the required scale is expensive and logistically demanding, and these systems generally exhibit weaker semantic reasoning and language understanding compared to LLM-based counterparts.

7. Conclusion

LLM's and VLA's have fundamentally altered the trajectory of robotics. Systems that were once rigid and task-specific are progressively evolving into general-purpose humanoids capable of understanding natural language, reasoning over complex goals,

and acting in unstructured real-world environments. However, significant limitations remain. Challenges related to physical grounding, inference latency, data requirements, and deployment reliability continue to constrain how confidently these systems can be trusted in real-world settings. This makes clear that LLM's alone are insufficient as a complete foundation for robotic intelligence.

The most likely path forward is not purely LLM-driven architecture, nor a return to purely classical approaches, but rather a principled integration of complementary systems. LLM's bring reasoning capability and natural language understanding, VLA's and world action models provide perception-to-action mapping grounded in physical context, world models contribute predictive understanding of physical dynamics, and classical controllers deliver the precision and stability that neural policies alone cannot yet guarantee.

LLM's are not the final answer, but they have been the catalyst that made general-purpose humanoid intelligence a credible pursuit.

References

- [1] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R.J. Rosario, K. Jeffrey, S. Jesmonth, N.J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," *arXiv preprint arXiv:2204.01691*, 2022. <https://arxiv.org/abs/2204.01691>
- [2] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann, *et al.* , "PaLM: Scaling Language Modeling with Pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023. <https://arxiv.org/abs/2204.02311>
- [3] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, *et al.* , "Llama 2: Open Foundation and Fine-Tuned Chat Models," *arXiv preprint arXiv:2307.09288*, 2023. <https://arxiv.org/abs/2307.09288>
- [4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N.J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "RT-1: Robotics Transformer for Real-World Control at Scale," *Robotics: Science and Systems (RSS)*, 2023. <https://arxiv.org/abs/2212.06817>
- [5] D. Driess, F. Xia, M.S.M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahber, 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," *International Conference on Machine Learning (ICML)*, 2023. <https://arxiv.org/abs/2303.03378>
- [6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M.G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N.J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W.E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, and M. Yan, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," *arXiv preprint arXiv:2307.15818*, 2023. <https://arxiv.org/abs/2307.15818>

- [7] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. 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," *arXiv preprint arXiv:2406.09246*, 2024. <https://arxiv.org/abs/2406.09246>
- [8] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusber, L. Groom, K. Hausman, B. Ichter, S. James, K. Kirmani, D. Lee, Y. Lee, L. Lee, S. Levine, A. Ly, H. Michalewski, I. Mordatch, K. Pertsch, L. Pinto, A. Sawhney, J. Tompson, and P. Xu, " π_0 : A Vision-Language-Action Flow Model for General Robot Control," *arXiv preprint arXiv:2410.24164*, 2024. <https://arxiv.org/abs/2410.24164>
- [9] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, Y.L. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An Open-Source Generalist Robot Policy," *arXiv preprint arXiv:2405.12213*, 2024. <https://arxiv.org/abs/2405.12213>
- [10] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid Loss for Language Image Pre-Training," *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. <https://arxiv.org/abs/2303.15343>
- [11] M. Oquab, T. Darcet, T. Moutakanni, H.V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning Robust Visual Features without Supervision," *Transactions on Machine Learning Research (TMLR)*, 2024. <https://arxiv.org/abs/2304.07193>
- [12] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, P. Sermanet, N. Brown, T. Jackson, L. Luu, S. Levine, K. Hausman, and B. Ichter, "Inner Monologue: Embodied Reasoning through Planning with Language Models," *Conference on Robot Learning (CoRL)*, 2023. <https://arxiv.org/abs/2207.05608>
- [13] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents," *International Conference on Machine Learning (ICML)*, 2022. <https://arxiv.org/abs/2201.07207>
- [14] Open X-Embodiment Collaboration, "Open X-Embodiment: Robotic Learning Datasets and RT-X Models," *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. <https://arxiv.org/abs/2310.08864>
- [15] Y. LeCun, "A Path Towards Autonomous Machine Intelligence," *OpenReview preprint*, version 0.9.2, 2022. <https://openreview.net/pdf?id=BZ5a1r-kVsf>

About the Authors



Banuru Krishna Chaitanya Reddy works as specialist programmer(AI/ML). He has 2.5 years of experience working with Infosys across research, AI product development for various clients. Has experience building projects from ground up to deployments at scale.

For more information, contact askus@infosys.com

Infosys
Navigate your next

© 2026 Infosys Limited, Bengaluru, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.