AI

Review

Integrating Large Language Models into Robotic Autonomy: A Review of Motion, Voice, and Training Pipelines

Yutong Liu, Qingquan Sun * and Dhruvi Rajeshkumar Kapadia

School of Computer Science and Engineering, California State University San Bernardino,
San Bernardino, CA 92407, USA; yutong.liu@csusb.edu (Y.L.); 008474096@coyote.csusb.edu (D.R.K.)
* Correspondence: qsun@csusb.edu

Abstract

This survey provides a comprehensive review of the integration of large language models (LLMs) into autonomous robotic systems, organized around four key pillars: locomotion, navigation, manipulation, and voice-based interaction. We examine how LLMs enhance robotic autonomy by translating high-level natural language commands into low-level control signals, supporting semantic planning and enabling adaptive execution. Systems like SayTap improve gait stability through LLM-generated contact patterns, while TrustNavGPT achieves a 5.7% word error rate (WER) under noisy voice-guided conditions by modeling user uncertainty. Frameworks such as MapGPT, LLM-Planner, and 3D-LOTUS++ integrate multi-modal data—including vision, speech, and proprioception—for robust planning and real-time recovery. We also highlight the use of physics-informed neural networks (PINNs) to model object deformation and support precision in contact-rich manipulation tasks. To bridge the gap between simulation and real-world deployment, we synthesize best practices from benchmark datasets (e.g., RH20T, Open X-Embodiment) and training pipelines designed for one-shot imitation learning and cross-embodiment generalization. Additionally, we analyze deployment trade-offs across cloud, edge, and hybrid architectures, emphasizing latency, scalability, and privacy. The survey concludes with a multi-dimensional taxonomy and cross-domain synthesis, offering design insights and future directions for building intelligent, human-aligned robotic systems powered by LLMs.

Keywords: large language models (LLMs); autonomous navigation; simulation-to-real transfer; multi-modal datasets; voice-based interaction; Task and Motion Planning (TAMP); physics-informed neural networks (PINNs); semantic reasoning; robot manipulation; reinforcement learning; human–robot interaction (HRI); cloud-edge hybrid architecture

check for updates

Academic Editor: Gianni D'Angelo

Received: 7 April 2025
Revised: 29 May 2025
Accepted: 5 June 2025
Published: 15 July 2025

Citation: Liu, Y.; Sun, Q.; Kapadia, D.R. Integrating Large Language Models into Robotic Autonomy: A Review of Motion, Voice, and Training Pipelines. *AI* **2025**, *6*, 158. https://doi.org/10.3390/ai6070158

1. Introduction and Motivation

Robotics has evolved significantly from early automata and programmable industrial machinery to highly sophisticated systems with advanced capabilities. A pivotal moment in this evolution was the introduction of Unimate, the first industrial robot, which began operating on a General Motors assembly line in 1961 (see Figure 1). Unimate revolutionized manufacturing by automating repetitive and hazardous tasks, laying the groundwork for modern industrial robotics [1]. Over time, robotics diversified into multiple categories, including industrial robots for precision manufacturing, service robots for household and professional assistance, medical robots for surgical precision, mobile robots for dynamic navigation, aerial robots for environmental monitoring, and social robots for empathetic human interaction [1,2]. This growth has been largely driven by advancements in Artificial Intelligence (AI), Machine Learning (ML), and reinforcement learning (RL), which have collectively improved robots' ability to learn, adapt, and operate autonomously across diverse environments [1,3].

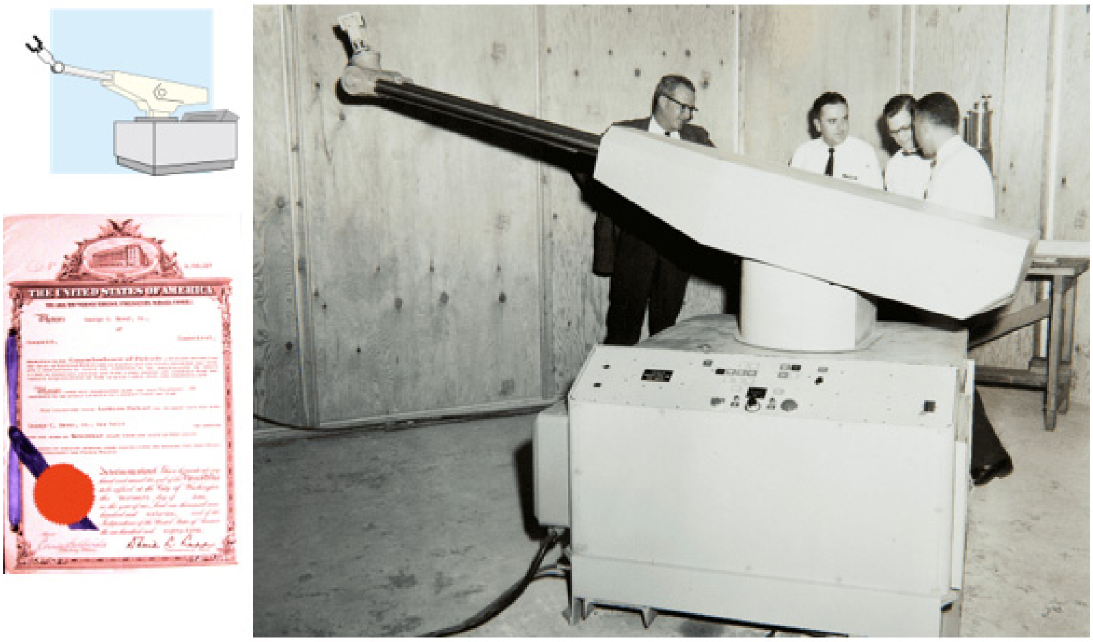


Figure 1. Unimate was the first industrial robot, which worked on a General Motors assembly line at the Inland Fisher Guide Plant in Ewing Township, New Jersey, in 1961.

The integration of robotics into daily life now transcends traditional industrial use cases, extending into university campuses, homes, and workplaces, where autonomous robots perform tasks such as navigation, classroom assistance, campus security, and household services [4–6]. At the core of this transformation lies robot autonomy, defined as the ability of a robot to perceive its surroundings, process sensory data, make informed decisions, and execute tasks without continuous human intervention [1,7]. Autonomy relies on critical elements such as sensor fusion, motion planning, and decision-making algorithms, which enable robots to function effectively in dynamic, unpredictable environments [5].

However, the increasing complexity of tasks assigned to autonomous robots requires advancements not only in hardware but also in intelligent software systems capable of high-level reasoning and decision-making. Recent developments in foundation models, particularly large language models (LLMs), Large Vision–Language Models (VLMs), and Large Audio-Language Models (ALMs), represent a transformative shift in robotics [2,8]. These models, pretrained on massive internet-scale datasets, exhibit exceptional capabilities in semantic reasoning, natural language understanding, and cross-modal integration, making them invaluable for tasks requiring contextual awareness and adaptable behavior [2,6,8]

Unlike traditional task-specific models, foundation models demonstrate zero-shot learning abilities, allowing robots to generalize knowledge across unseen tasks without requiring extensive retraining. Through effective task decomposition, these models enable robots to break down high-level user commands into executable actions, improving their ability to manage long-horizon tasks and respond dynamically to environmental changes [2,6]. Additionally, foundation models enhance robot capabilities in environmental reasoning, object manipulation, and path optimization, bridging critical gaps between perception, planning, and control.

Yet, despite these advancements, deploying foundation models in real-world robotics scenarios presents significant challenges. Factors such as data scarcity, high variability in operating environments, and real-time performance constraints pose ongoing hurdles [9–11]. Addressing these obstacles requires fine-tuning foundation models for robotics-specific tasks, optimizing their inference speed, and developing scalable datasets that bridge the simulation-to-reality (sim-to-real) gap.

In addition to foundation models, the integration of reinforcement learning (RL) and deep reinforcement learning (DRL) has played a pivotal role in improving robot adaptability. While RL allows agents to learn optimal behaviors through trial-and-error interactions with their environment, DRL combines RL principles with deep neural networks to address high-dimensional state spaces and dynamic environments. Algorithms such as Deep Q-Networks (DQNs) and Proximal Policy Optimization (PPO) have significantly improved robot locomotion and adaptability across uneven and unpredictable terrains [12]. However, traditional RL approaches often suffer from sample inefficiency, reward misalignment, and limited generalization capabilities [5,13]. To address these challenges, LLM-enhanced RL has emerged as a promising paradigm, leveraging the contextual reasoning abilities of LLMs to improve sample efficiency, reward function design, and policy generalization [3].

Alongside these advancements, voice-based interaction has gained prominence as an intuitive interface for human–robot collaboration. Voice command integration, combined with LLMs for semantic interpretation and reasoning, significantly enhances robot usability and responsiveness. This approach allows robots to understand nuanced verbal instructions, adapt to diverse communication styles, and offer interactive support in domains such as campus guidance, household assistance, and security patrol operations [12].

Despite the progress made, achieving robust robot autonomy continues to face persistent challenges, including environmental uncertainty, real-time decision-making bottlenecks, and the need for continuous learning and adaptation [5,14]. Addressing these issues will require advancements in high-level planning, low-level motor execution, error recovery mechanisms, and safety evaluation protocols. Moreover, improving reproducibility and standardizing benchmarks for evaluating robotic systems are critical steps in advancing the field.

This survey aims to provide a comprehensive overview of the state-of-the-art research on LLM-augmented robotics, model-enhanced RL frameworks, and voice-based integration systems. By synthesizing recent insights from multiple studies [2,3,15–18], we aim to identify emerging trends, address persistent challenges, and propose future opportunities for developing intelligent, adaptable, and multi-functional robots. Our focus lies in exploring core competencies such as locomotion, manipulation, environment modeling, and multi-modal integration, with the ultimate goal of contributing to the creation of versatile robots capable of supporting classroom technology, facilitating campus tours, and assisting with household tasks.

2. Taxonomy Robot Competencies

To systematically analyze how LLMs enhance robotic autonomy, we propose a multi-dimensional taxonomy of robot competencies. This framework classifies competencies along three key dimensions: (1) task type—what the robot is doing, (2) integration method—how LLMs are embedded into control systems, and (3) input modality—how users communicate with the robot. This structured lens enables researchers to evaluate current LLM-robotic systems and identify design trade-offs, technical challenges, and deployment opportunities.

As demonstrated in Table 1, robotic tasks can be grouped into four categories: locomotion, navigation and mapping, manipulation, and HRI (human-robot interaction). Each category showcases different LLM roles, from converting intent into control specs to affordance-based planning and dialogue management. This classification helps map LLM functionality to diverse real-world deployment scenarios, such as campus navigation, security patrols, or classroom assistance.

Table 1. Dimension 1: task type.

Task Category	Examples	LLM Role	Deployment Scenarios
Locomotion	Walking, obstacle avoidance	Translate intent into control specs	Campus navigation, terrain traversal
Navigation and Mapping	Path planning, localization	Semantic decomposition of routes	Delivery, tour guides, security patrols
Manipulation	Object grasping, assembly, sorting	Affordance-based task planning	Labs, classrooms, domestic service
HRI and Interaction	Conversations, assistance, instruction	Dialogue management, context memory	Reception, academic support, elderly care

Table 2 outlines how LLMs can be embedded into control systems across different integration layers. High-level reasoning provides flexibility but suffers from slower response time, while low-level embedding enables faster execution with safety risks. Hybrid approaches aim to balance these trade-offs by integrating LLMs with Simultaneous Localization and Mapping (SLAM) or Deep Reinforcement Learning (DRL) methods.

Table 2. Dimension 2: integration method.

Integration Type	Description	Design Trade-Offs	Challenges
High-Level Reasoning	LLMs generate plans, tasks, or code	Flexible, reusable, easy to prompt	Slower response, hallucination risk
Mid-Level API	LLMs convert commands into API calls or parameters	Transparent, debuggable, modular	Requires predefined API mapping
Low-Level Embedding	LLM outputs directly drive control signals	Fast, continuous control possible	Safety risks, poor generalization
Hybrid Hierarchical	Combines LLM planning with DRL/SLAM execution	Balanced adaptability and control	Complex architecture, latency handing

Table 3 categorizes how input modalities influence communication between humans and robots. Natural language interfaces benefit from parsing and reasoning capabilities, while visual, haptic, and multi-modal inputs enable perception alignment and richer interactions. These modalities introduce distinct challenges, from noise and ambiguity to bandwidth limitations.

Table 3. Dimension 3: input modality.

Modality	Use Case	LLM Utility	Constraints
Natural Language	Speech, typed text	Parsing, grounding, semantic reasoning	Noise, ambiguity, multilinguality
Visual Inputs	Image, video, environment maps	Scene understanding, VLM grounding	Vision–language alignment
Haptic/Sensor	Touch, IMUs, LIDAR	Contextual feedback for manipulation or movement	Sensor fusion complexity
Multi-Modal Fusion	Any combination above	Enables holistic perception and reasoning	Synchronization and bandwidth

To bridge the taxonomy with practical application, Table 4 summarizes how specific competencies—like adaptive locomotion, voice interaction, and contextual planning—map to LLM contributions and corresponding deployment scenarios.

Table 4. Mapping competencies to deployment scenarios.

Competency	LLM Contribution	Deployment Scenarios
Adaptive Locomotion	Gait planning, obstacle context reasoning	Campus robots climbing stairs or avoiding clutter
Voice Interaction	Natural language understanding and instruction	User-guided assistance or Q&A for new students
Contextual Planning	Multi-step task graph generation	Rearranging lab tools, preparing classrooms

This taxonomy provides a conceptual scaffold to understand how LLMs enhance various dimensions of robotic autonomy. By systematically organizing task types, integration methods, and input modalities—alongside their respective trade-offs, technical challenges, and deployment contexts—we offer a structured lens for analyzing current systems and identifying future opportunities. This framework is intended not only to classify existing research but also to inform the design of next-generation robots that are adaptive, multimodal, and context-aware. As the field progresses, extending this taxonomy to include evaluation benchmarks, ethical implications, and safety protocols will be vital for ensuring responsible and robust deployment of LLM-driven robotics.

3. LLMs in Robotic Autonomy: Locomotion, Navigation, and Manipulation with Voice-Based Interaction

Modern large language models (LLMs) have ushered in a transformative era in robotic autonomy—spanning from low-level control, such as generating foot contact patterns for quadrupeds, to high-level planning tasks like multi-step code generation [1,2,18–20]. LLMs can be adapted to diverse robotic competencies, including a reward design for reinforcement learning, low-level actuator commands, complex multi-step manipulation, scene understanding, and planning that integrates knowledge from vision–language models [9,21,22]. Despite varying in their primary focus—some concentrate on real-time gait control and others on elaborate loco-manipulation sequences—these approaches share a unifying theme: harnessing natural language as a flexible interface for interpreting instructions, guiding planning, and adapting behaviors in dynamic environments.

As illustrated in Figure 2, LLMs function as translators between user commands (e.g., "Turn slightly left, take a photo, and give me the apple on the table") and low-level control policies. High-level instructions are decomposed into API calls, semantic parameters, or subtask graphs, which are then interpreted by reinforcement learning (RL) controllers to produce real-time torque or trajectory commands. This layered pipeline enables diverse robotic embodiments—including humanoids, manipulators, and quadrupeds—to execute tasks with contextual awareness, physical precision, and autonomy in unstructured settings. Beyond enhancing task generalization, this architecture allows for intuitive voice interaction and more human-aligned robotic behavior.

One recent advance, TrustNavGPT [20], incorporates vocal affective cues—such as pitch, loudness, and speech rate—to model user uncertainty during spoken navigation commands. Unlike scene-graph-based systems like SayNav [21], which depend on static spatial representations, TrustNavGPT adapts in real time to ambiguous or noisy speech environments. It achieved over 80% task success in audio-guided simulations and real-world deployments, demonstrating robust resistance to adversarial audio input and offering a practical route to more trustworthy, voice-driven robotic navigation.

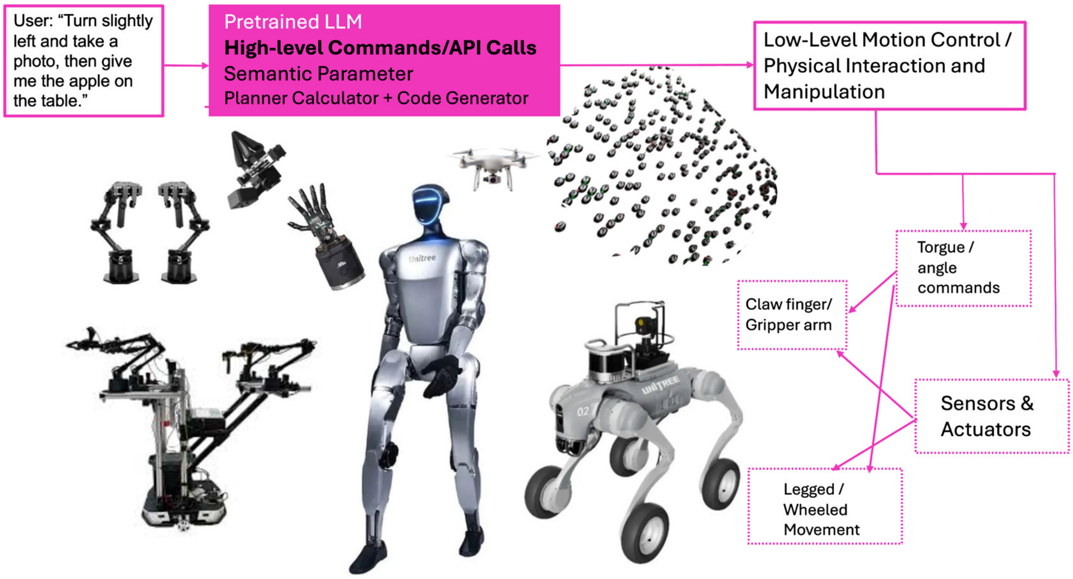

User: "Turn slightly left and take a photo, then give me the apple on the table."
Pretrained LLM
High-level Commands/API Calls
Semantic Parameter
Planner Calculator + Code Generator
Low-Level Motion Control / Physical Interaction and Manipulation
Torgue / angle commands
Claw finger/ Gripper arm
Sensors & Actuators
Legged / Wheeled Movement

Figure 2. Integration of pretrained large language models (LLMs) with low-level reinforcement learning (RL) controllers for robotic autonomy. The LLM interprets user instructions into semantic parameters and API calls, which are translated into torque or angle commands for various robot types (arms, humanoids, quadrupeds) via RL control policies.

This section presents a synthesis of how LLMs are actively enabling three key pillars of embodied intelligence: (1) locomotion, where models like SayTap [19] and WildLMa [17] convert verbal intent into foot contact patterns or terrain-aware motion plans; (2) navigation and semantic planning, where frameworks such as MapGPT [23], SayNav, and TrustNavGPT demonstrate real-time scene-aware reasoning and long-horizon planning; and (3) physical interaction and manipulation, where frameworks like LLM+A [24] and BETR-XP-LLM [25] guide robots through affordance reasoning and error-resilient control. Finally, voice-based interaction as a cross-domain interface addresses the growing role of speech as a natural and unified input modality across all three domains, highlighting architectures, speech-to-text pipelines, and fallback control strategies. The next section builds on this integrated view by organizing these advancements into a functional taxonomy of robotic competencies.

3.1. LLMs in Locomotion Control

3.1.1. Bridging Natural Language to Low-Level Motion

At the core of this interface lies low-level motion control, which refers to the direct management of a robot's actuators, including joint angles, torques, motor velocities, and contact timings. These are the physical control signals that govern how limbs move, how much force to apply, and how fast to execute a motion. Translating high-level language like "walk slowly to the left and grasp the handle" into these low-level commands requires an intermediate reasoning system—typically a policy model such as reinforcement learning (RL)—to decode verbal instructions into precise motion primitives. NLP thus enables robots to bypass hand-coded control scripts and instead generate adaptable behaviors directly from human intent.

Natural language processing (NLP) serves as a critical bridge between high-level human commands and low-level robotic control signals, enabling intuitive and precise robotic behavior execution. Traditional interaction methods, such as tactile indicators and gesture-based control, often require users to undergo specific training or rely on predefined movement patterns to guide robot behavior [26]. In contrast, natural language (NL)-based execution (NLexe) offers an intuitive, flexible, and user-friendly interface, allowing even non-experts to interact seamlessly with robots [7,27].

One of the key strengths of NL is its ability to deliver precise execution requests, including specifications for actions, speed, tools, and locations, without relying on complex gesture or pose recognition systems [12,28]. Standard linguistic structures, such as those found in English, Chinese, and German, inherently support diverse command variations, eliminating the need for custom-designed execution patterns [29,30]. This efficiency not only simplifies robot interaction but also enables dynamic task adaptation in unstructured environments [18,30–32].

For low-level motion control, NLP decodes high-level verbal instructions (e.g., “Move forward cautiously and place the object on the left table”) into structured parameters such as joint angles, torque commands, and timing patterns. These parameters are then enforced by reinforcement learning (RL) controllers, ensuring stable execution and real-time adaptability. Moreover, NLP frees users from physical interaction constraints, allowing hands-free task delegation, such as saying “Hold the handle and open the door” without needing to physically guide the robot.

The importance of NLP in robotic motion control extends across domains, including daily assistance, healthcare automation, manufacturing workflows, and navigation planning [32]. By serving as a natural interface for precise command translation, NLP empowers robots to interpret nuanced instructions, dynamically adjust control signals, and execute tasks efficiently. As shown in Figure 3, the integration of NLP with low-level control systems creates a cohesive pipeline—from human instruction parsing via NLP to real-time execution through RL controllers—enabling robots to seamlessly translate natural commands into actionable, fine-grained behaviors on physical hardware [19].

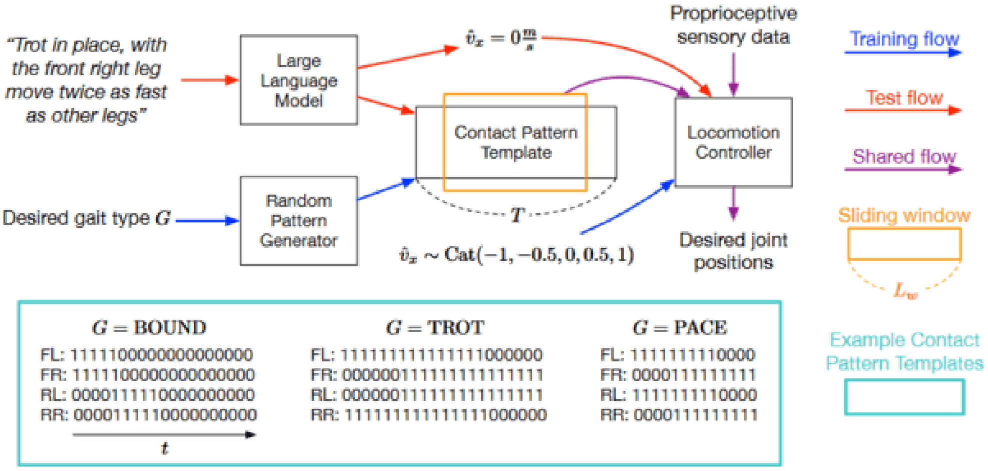


Figure 3. An overview of the SayTap-style pipeline. Human commands are translated by the LLM into a contact pattern template (rows of zeros/ones), which the DRL controller uses to produce stable gait behaviors. During training (blue arrows), patterns are randomly sampled; during testing (red arrows), the system relies on user commands interpreted by the LLM [19].

3.1.2. Long-Horizon Reasoning for Loco-Manipulation

Long-horizon reasoning in loco-manipulation refers to a robot's capacity to plan, coordinate, and execute complex, multi-step tasks over extended durations in dynamic, partially observable environments. Unlike low-level motor tasks such as walking, stepping over obstacles, or adjusting balance, long-horizon tasks demand the integration of semantic understanding, high-level intent parsing, and real-time adaptability. These capabilities are especially critical for service robots operating in unstructured settings such as university campuses, hospitals, or homes.

Consider, for example, a campus service robot instructed to "turn off all classroom lights before exiting the building". This task entails multiple steps: navigating to each classroom, locating the switch, assessing reachability, climbing an object if needed, executing the switch press, and then proceeding to the next location or exit. Completing such a task requires more than navigation—it necessitates contextual reasoning (e.g., interpreting "lights" and "exit"), spatial and temporal memory, and robust motor control in response to unpredictable environmental constraints such as the presence or absence of furniture, obstacles, or human traffic [9,28,33–37].

At the architectural level, these systems often rely on a hierarchical planning–control framework. Large language models (LLMs) are employed at the high-level reasoning layer to parse instructions, decompose them into atomic or skill-level subtasks, resolve ambiguities, and calculate parameters such as estimated reach, object location, or optimal sequence [24,38]. These task decompositions are passed down to low-level controllers—often based on reinforcement learning (RL) or Behavior Cloning—which manage closed-loop actuation based on real-time sensory feedback. This decoupling of symbolic reasoning from reactive execution improves robustness, reduces computational load during real-time operation, and allows for better modularity and the reuse of learned skills.

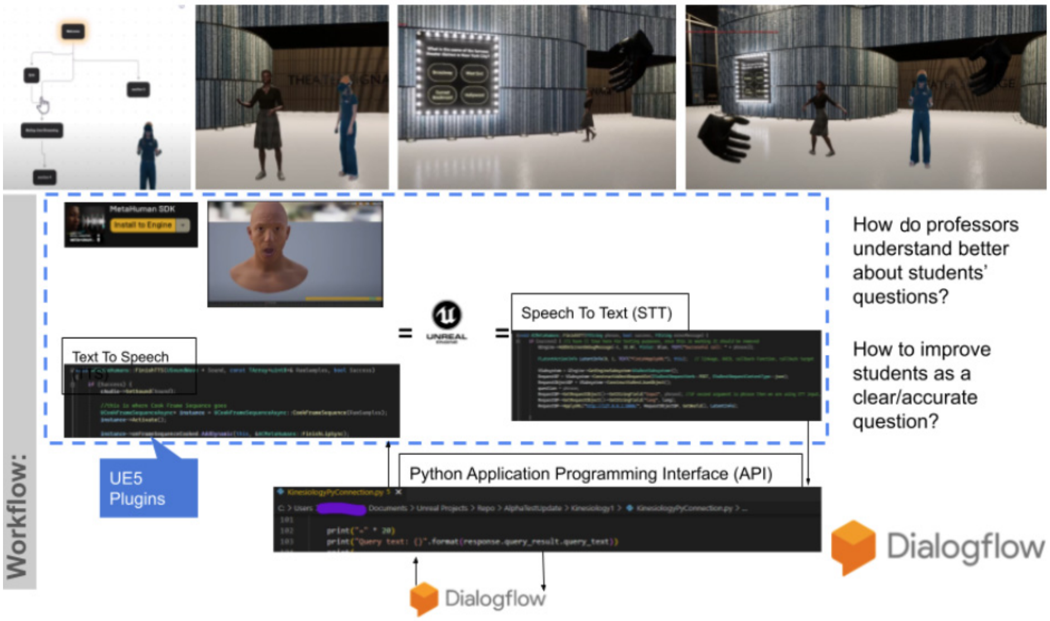
Recent LLM-enhanced systems such as SayTap [19] and LLM-Planner demonstrate the benefits of long-horizon skill coordination. SayTap, for instance, uses LLM-generated binary contact cues to guide locomotion across sequential contact tasks, effectively embedding planning priors into physical action sequences. Similarly, LLM-Planner dynamically adjusts execution plans based on multi-modal feedback, enabling robots to revise their strategy when encountering novel layouts or human interruptions.

In practical deployments, long-horizon reasoning empowers robots to adapt in real time to dynamic scenes. In campus logistics, this might involve rerouting to avoid a crowded hallway, or recalibrating a manipulation sequence to accommodate different desk heights or lighting conditions. By leveraging LLMs' pretrained knowledge and contextual reasoning, these systems can construct dynamic task graphs that account for spatial relationships, physical feasibility, and task priorities—resulting in more efficient and reliable performance in complex environments [38–40].

3.2. LLMs in Navigation and Semantic Planning

Integrating large language models (LLMs) into robotic navigation frameworks has transformed how robots understand and execute spatial tasks. Traditional algorithms such as A* or RRT depend on geometric heuristics and fixed maps, often failing in dynamic, ambiguous environments [41,42]. In contrast, LLM-based systems combine semantic reasoning with adaptive path execution, enabling real-time response to user intent, spatial context, and unexpected changes. These advancements support high-level task decomposition, dynamic error recovery, and long-horizon planning—essential for next-generation autonomous agents operating in complex environments [43].

LLMs interpret human instructions like "Go to the third-floor kitchen avoiding the construction area" by mapping abstract goals into executable way-point paths and control sequences. This capability is enhanced when paired with visual or topological representations, allowing LLMs to reason across space, time, and interaction intent. Figure 4 demonstrates such integration in a virtual agent scenario, where voice commands guide a simulated professor character through interactive navigation.


Workflow:
MetaHuman SDK
Install to Engine
Text To Speech
UNREAL ENGINE
Speech To Text (STT)
UE5 Plugins
Python Application Programming Interface (API)
Dialogflow
Dialogflow
How do professors understand better about students' questions?
How to improve students as a clear/accurate question?

KinesiologyPyConnection.py 5 ×

C: > Users [redacted] Documents > Unreal Projects > Repo > AlphaTestUpdate > Kinesiology1 > KinesiologyPyConnection.py > ...

```
print("-" * 20)
print("Query text: {}".format(response.query_result.query_text))
```

Figure 4. The system integrates Text-to-Speech (TTS) and speech-to-text (STT) functionalities via a custom Python APIs, enabling seamless natural language interaction with virtual characters, such as Professor Papailler. This setup allows users to receive tailored guidance, whether through educational walkthroughs, dance motion capture demonstrations, or knowledge-driven conversations, demonstrating real-time adaptability in response-to-user inputs.

3.2.1. High-Level Planning and Decomposition

High-level semantic planning is the process by which robots interpret abstract human instructions—such as "go to the library avoiding construction zones"—and transform them into structured sequences of actions that align with task goals and environmental constraints. This form of planning relies on semantic understanding, which goes beyond simple geometric pathfinding to incorporate task intent, contextual relevance, and environmental dynamics. Large language models (LLMs) play a key role here by mapping natural language to meaningful sub-goals, spatial relations, and behavioral priorities.

Closely tied to semantic planning is the concept of high-level task decomposition—the process of breaking down a complex instruction into manageable, sequential subtasks or atomic actions. While planning defines the overall route or strategy, decomposition structures that plan into actionable steps. For instance, the command "deliver this package to Room 205 and return to the charging station" would be decomposed into (1) locate package, (2) navigate to Room 205, (3) confirm delivery, and (4) return to base. This hierarchy supports both symbolic reasoning and low-level control integration.

In LLM-enabled navigation frameworks, these two processes often work hand in hand. MapGPT, for example, uses map-guided prompting to perform semantic planning at a topological level—generating global strategies for traversing large environments [23]. Meanwhile, SayNav incrementally constructs 3D scene graphs to adaptively re-plan at a local scale, enabling responsiveness in cluttered or dynamic conditions [21]. Both systems reflect the dual role of high-level reasoning: establishing long-range intent while supporting decomposition into local, context-aware behaviors.

Frameworks like LLM-Planner [39] advance this integration by using multi-modal feedback—including vision, language, and proprioception—to continuously revise both plans and subtasks in real time. This allows robots to remain goal-aligned even when faced with unexpected changes or interruptions, achieving more robust and adaptable autonomy in real-world environments.

3.2.2. Dynamic Execution and Recovery

Dynamic execution and recovery are essential for robust robotic autonomy, especially in real-world environments where uncertainty is inevitable. Unlike static planning, which assumes that tasks will proceed exactly as expected, dynamic execution allows robots to interpret sensor input, adapt to changes, and make real-time decisions during task execution. Recovery, in this context, refers to a robot's ability to detect failures—such as falling, misalignment, or command ambiguity—and then autonomously revise its behavior to continue working toward its goal.

For instance, in human–robot interaction, systems like TrustNavGPT enhance resilience by analyzing vocal cues such as pitch or speech tempo to infer user uncertainty. When a command is unclear or ambiguous, the robot can prompt clarification or re-plan its route instead of stalling. Similarly, VoicePilot demonstrates real-time adjustment by parsing natural commands with temporal modifiers—such as "go faster" or "wait a moment"—through LLMs like GPT-3.5 Turbo and executing behavior that aligns with the user's intent [20].

These capabilities are especially important when physical or environmental conditions change mid-task. A quadruped robot like Unitree Go2 may experience initial posture misalignment not only due to uneven terrain but also from unpredictable user interactions—such as being powered on in awkward orientations or operated without following startup instructions. Without the ability to recover from this state—by recalibrating joint positions or shifting weight distribution—it risks falling or failing to start altogether. Similarly, in manipulation tasks, a robot grasping an object might misjudge the grip force or object geometry. Frameworks like BETR-XP-LLM [25] respond by adjusting behavior trees on the fly, retrying with alternative strategies or force parameters to complete the action successfully.

Even high-performing models like RT-1 [14], though capable of impressive large-scale skill generalization, struggle with such real-time adaptability due to their static architecture. In contrast, lightweight LLM-integrated planners that continuously monitor feedback can respond immediately to failure or deviation—making them more suited to deployment in open-ended, dynamic environments.

In short, dynamic execution and recovery are not secondary features but foundational to functional autonomy. They enable robots not only to act but to reflect, adapt, and persist—much like humans—when things do not go as planned.

3.3. LLMs in Physical Interaction and Manipulation

Large language models (LLMs) are reshaping robotic manipulation by enabling systems to interpret natural language instructions and translate them into structured, context-aware behaviors. Manipulation—the ability to grasp, push, align, or assemble objects—relies not only on semantic understanding but also on the robot's capacity to predict and respond to physical interactions. LLMs provide the reasoning layer: they parse high-level instructions, identify task goals, and decompose actions into steps. Yet to perform reliably in the real world—where object deformation, contact dynamics, and environmental uncertainty are constant—semantic reasoning alone is not enough. This is where physics-informed neural networks (PINNs) play a crucial role [44].

Physics-informed neural networks (PINNs) enhance manipulation robustness by embedding physical constraints—like elasticity, friction, and conservation laws—directly into learning architectures. These models reduce the dependency on large datasets by integrating prior knowledge, making them ideal for environments with limited or noisy data. For instance, PINN-Ray accurately predicts the deformation behavior of Fin Ray grippers under varying loads, enabling soft robotic hands to interact more precisely with fragile or irregularly shaped objects [45]. Similarly, recent studies [46] have shown that combining physics-informed neural networks with reinforcement learning significantly improves the trajectory tracking performance in robotic arms, especially when manipulating deformable or non-rigid objects.

In manipulation tasks, where contact forces and object responses are difficult to model explicitly, PINNs act as a physical interpreter that refines low-level execution, complementing the LLM's high-level intent [44]. When integrated into the robotic stack, PINNs support adaptive execution—modifying the grip force, adjusting alignment, or compensating for unexpected resistance—thereby reducing failure rates and improving recovery.

Frameworks such as BETR-XP-LLM and CriticGPT demonstrate the power of coupling semantic reasoning with real-time feedback. When a flexible object cannot be inserted into a tight slot, BETR-XP-LLM reconfigures its behavior tree to attempt an adjusted strategy. With the addition of PINNs, such adaptations can be guided by physically plausible estimations of force, deformation, and compliance, rather than trial and error. In collaborative scenarios, human–robot collaboration (HRC) systems can further refine control by integrating human corrections, while still maintaining the semantic coherence provided by the LLM layer [47].

In sum, the fusion of LLMs for semantic reasoning and PINNs for physically informed adaptation enables robots to manipulate the world with both symbolic understanding and physical realism. This hybrid architecture represents a significant leap toward robust, generalizable, and human-aligned robotic manipulation. As demonstrated in Table 5, several state-of-the-art frameworks highlight different approaches to integrating LLMs into robotic systems. Table 1 summarizes their primary domains, core focuses, and key strengths, while also identifying trade-offs and deployment challenges. This comparative overview provides essential context for understanding the landscape of LLM-enhanced autonomy and guides future development efforts.

Table 5. Comparative Analysis of LLM-Based Robotic Frameworks Across Domains, Strengths, and Deployment Challenges.

Framework	Primary Domain	Focus	Key Strengths	Trade-Off Deployment Challenges
RT-1 [14]	Vision-based manipulation	Web-scale training for multi-modal reasoning	Generalization across visual tasks using LLM + VLM	Requires massive training data and limited in real-time planning
SayTap [19]	Locomotion (quadruped)	Language to quadruped locomotion control	Simple, low-level control via foot contact patterns	Limited to locomotion and requires careful reward shaping
TrustNavGPT [20]	Voice-Guided Navigation	Navigation with uncertainty modeling from STT inputs	Chain-of-thought parsing, low WER (5.7%)	Requires hybrid cloud-edge setup for robust performance
SayNav [21]	Navigation in dynamic spaces	Multi-object navigation in dynamic scenes	Real-time scene graph update, fast local adaption	Limited global path planning, mainly simulation
MapGPT [23]	Vision–Language Navigation	Vision–Language Navigation (VLN) with topological maps	Strong global planning, multi-step reasoning	High reliance on map quality and lacks direct low-level control
BETR-XP-LLM [25]	Loco-manipulation	Long-horizon loco-manipulation	Dynamic behavior tree generation, error recovery	High computational resource demand, rich skill library needed
LLM-Planner [38]	Loco-manipulation	Multi-modal hierarchical planning	Adaptively revises plans in dynamic scenes	Complex architecture and requires high-quality feedback integration
VoicePilot [43]	Voice interaction	Multi-step natural language control for assistive robots	Modifier-aware parsing, GPT-3.5-based semantic control	Cloud inference delay, privacy concern, and needs fallback control

3.4. Voice-Based Interaction as a Cross-Domain Interface

Designing robots that can learn from and interact with human speech in real-world environments is a complex but critical challenge. Voice input offers natural, hands-free command capabilities, enabling users to guide robots through tasks such as object retrieval, route navigation, and general assistance without the need for handheld interfaces. For instance, a user might say, “Could you help me find the dishwasher?” And the robot, through large language model (LLM)-enabled reasoning, would parse the command, infer movement paths, and confirm the relevant objects in its environment. Compared to traditional interfaces such as joysticks or keyboards, voice commands reduce the cognitive load on users and enable hands-free operation, which is particularly important in scenarios requiring quick or context-aware decision-making, such as robot-assisted navigation or task execution [42]. Furthermore, voice commands enhance accessibility for individuals with limited mobility, allowing them to interact with robots effectively [48].

Mobile control integration, on the other hand, serves as both a complementary and fallback mechanism for voice-based systems. Mobile apps enable remote teleoperation, providing users with a graphical interface to fine-tune robot movements or issue commands in noisy or ambiguous environments where voice commands may fail [49,50]. For instance, in the case of Unitree quadruped robots, mobile apps are often used for teleoperation and advanced control, offering flexibility for tasks such as autonomous navigation, environmental exploration, and real-time monitoring. Integration with mobile apps also enhances situational awareness by offering live camera feeds, object recognition data, and detailed diagnostic information, ensuring a robust and user-friendly robot operation framework.

3.4.1. Popular Architectures for LLM-Based Voice Command Systems

Voice-based interfaces have become a critical component of real-world robotic autonomy, enabling intuitive, hands-free command execution across locomotion, navigation, and manipulation tasks. In practical settings such as homes, campuses, or healthcare environments, spoken commands are often the most natural and accessible way for humans to interact with robots. However, real-life speech interaction is subject to a range of challenges—including background noise, ambiguous phrasing, speech variability, and dynamic environments—which can compromise reliability and responsiveness.

To mitigate these issues, modern systems frequently incorporate mobile control applications or physical controllers as fallback mechanisms. These tools allow users to visually confirm robot behavior, fine-tune movement, or override ambiguous commands in real time. In the case of quadruped robots like Unitree, mobile apps provide an essential interface for remote navigation, environmental perception, and task debugging [51–53]. Furthermore, emerging research is exploring the integration of virtual reality (VR) headsets and multi-modal interfaces to augment verbal commands with spatial cues or gestural input, enhancing situational awareness and user control.

These architectural strategies reflect a growing need to balance natural interaction with technical robustness, leading to diverse implementations across cloud, edge, and hybrid systems. The following section examines these architectures in detail, evaluating their trade-offs and suitability for different deployment scenarios [54,55].

Cloud-based architectures rely on remote servers to process speech to text (STT), natural language understanding (NLU), and command generation. They allow access to powerful LLMs and deep inference capabilities. Systems like TrustNavGPT use cloud-based models to integrate LLM reasoning with vocal affect detection, improving navigation in uncertain audio contexts. Similarly, VoicePilot uses GPT-3.5 Turbo to interpret multi-step commands. Other platforms such as IBM Watson Assistant, Baidu AI Cloud, and OpenAI Whisper provide scalable voice parsing and multilingual support. However, cloud systems require stable internet and may suffer from latency and data privacy concerns [56].

Edge-based architectures move all processing onto the robot's onboard hardware, enabling real-time performance and offline operation. These systems are ideal for privacy-sensitive or connectivity-limited environments. Examples include the Voice Recognition Robot, which uses Arduino boards for on-device STT [54], and interactive tour guide robots, which embed voice modules directly into their local controllers [52]. The main limitation is reduced computational capacity, which may hinder the understanding of complex or open-domain commands.

Hybrid architectures split computation between local devices and the cloud. For example, TrustNavGPT shifts between local and cloud-based processing depending on speech ambiguity. VoicePilot filters modifier terms locally (e.g., "gently", "faster") before deferring more complex reasoning to cloud-based LLMs. This hybrid design balances responsiveness with scalability and is well-suited for university campuses, smart homes, and rescue operations where network reliability varies. As summarized in Table 6, cloud, edge, and hybrid architectures each offer distinct trade-offs in latency, connectivity, and performance, shaping how LLM-powered robotic systems are deployed across environments such as smart campuses, mobile platforms, and assistive settings.

Table 6. The key characteristics of these architectural approaches are summarized.

Architecture	Latency	Connectivity	Strengths	Limitations	Ideal Use Cases	Example System
Cloud	Medium–High	High (Always Connected)	Scalable LLM access, rich semantic reasoning	Latency, privacy risks, dependency on internet	Smart campuses, labs, offices with stable WiFi	TrustNavGPT [20]
Edge	Low	None	Fast response, privacy preserving, offline-ready	Limited model size and complexity	Home healthcare, tour guide robot, mobile platforms	Voice Rec Robot [51]
Hybrid	Low–Medium	Intermittent	Balances performance and flexibility	Requires smart task delegation logic	Campus assistants, rescue robots, smart buildings	VoicePilot [43]

3.4.2. Reliable Techniques in LLM-Based Voice Control

Robust voice control requires a pipeline of dependable subsystems. First, STT systems like OpenAI Whisper and Google STT transcribe user input. Whisper achieves strong noise robustness with a 5.7% word error rate (WER), while Google STT offers low-latency performance and extensive multilingual support [23].

After transcription, semantic parsing is handled by LLMs such as GPT-3.5 Turbo, Gemini, or ChatGPT. These models support multi-step reasoning, disambiguation prompts, and context-aware modifiers. For example, TrustNavGPT employs chain-of-thought parsing, while VoicePilot parses semantic intent and command nuance.

To bridge language and physical interaction, object detection systems like YOLOv8 and RTAB-Map ground verbal commands into sensor-based perceptions. For instance, voice-directed robots in early education scenarios use YOLOv8 to identify colored blocks and respond to commands like "pick up the red cylinder" [55].

Finally, error recovery mechanisms are essential. Systems like TrustNavGPT prompt users to rephrase unclear input, while VoicePilot automatically switches to mobile control when ambiguity is detected. Table 7 provides a taxonomy of common failure modes and recovery strategies.

Table 7. Taxonomy of voice control failures and recovery strategies.

Failure Type	Cause	Recovery Mechanism	Example System
STT Error	Noise, accent mismatch	Retry prompt, confirmation query	Whisper
Semantic Ambiguity	Vague or compound instructions	Clarification prompt via LLM	TrustNavGPT
Object Grounding Error	Occlusion, detection mismatch	Visual feedback loop, retry	YOLOv8
LLM Hallucination	Overconfident command inference	Confidence threshold fallback, mobile app	VoicePilot

Table 8 offers a comparative overview of six leading cloud voice platforms based on latency, WER, environmental robustness, and customization potential. Services like Google

Dialogflow and Microsoft Azure offer fast, adaptive transcription with strong customization tools, while Whisper provides noise-resilient open access. In contrast, enterprise-grade platforms like IBM Watson deliver robust modeling with higher latency.

Table 8. Comparison of selected cloud-based voice recognition services.

Service	Latency	Accuracy (WER)	Noise Robustness	Customization	Strengths	Weaknesses
Google Dialogflow	Low	5.3%	High	Extensive	Real-time performance, multilingual, wide coverage	Requires task-specific tuning for optimal results
Amazon Transcribe	Medium	6.2%	Medium	Moderate	AWS integration, speaker separation, scalable	Expensive at scale
Microsoft Azure	Medium	5.5%	High	Extensive	Customizable acoustic, accent adaptation	Complex for smaller projects
IBM Watson	Medium	6.1%	Medium	High	Enterprise-grade, privacy and modeling	Higher latency and less suitable for real-time robotics
Baidu AI Cloud	Medium	5.8%	High	Moderate	Noise handling, support for Chinese dialects	Limited global adoption and integration
OpenAI Whisper	Medium	5.7%	High	Moderate	High transcription accuracy, noise tolerance	Limited customization

4. Training Frameworks, Datasets, and Sim-to-Real Deployment

The transition from theoretical advancements in large language models (LLMs) and reinforcement learning (RL) algorithms to practical robotic training and deployment relies heavily on robust training frameworks, reliable simulation platforms, and diverse datasets. Campus service robots, including tour guides, classroom assistants, academic support agents, and campus safety monitors, require adaptive algorithms and scalable models to function effectively in real-world scenarios. Techniques such as Proximal Policy Optimization (PPO) and hierarchical RL (HRL) enable robots to learn structured behaviors [4], while domain randomization and incremental fine-tuning bridge the gap between simulated and real-world environments. The integration of 2D visual data, 3D point clouds, and natural language instructions ensures robots can perceive and interact with their surroundings more effectively. Locally ongoing training, supported by edge computing and federated learning, allows robots to refine their capabilities in real time while maintaining data privacy and compliance with institutional IT policies [57].

Datasets play an essential role in driving this transition, acting as the foundation for developing robust and adaptable robotic systems. A diverse dataset exposes robots to a variety of tasks, environmental conditions, and object interactions, ensuring better generalization across unseen scenarios. For example, RH20T focuses on one-shot imitation learning through multi-modal sensory data, while Open X-Embodiment leverages cross-embodiment experiences to facilitate multi-robot adaptability. Similarly, BridgeData V2 provides goal- and language-conditioned demonstrations to enhance policy learning [31,58], and OpenLORIS-Object emphasizes object recognition tasks under dynamic and challenging conditions [49]. Beyond diversity, dataset fairness ensures that no systematic biases emerge, allowing equitable task execution across varying environmental setups and demographic interactions. Safety remains a central focus, with datasets incorporating failure cases, obstacle avoidance protocols, and emergency scenarios to ensure robust deployment in sensitive campus environments.

A scalable campus robotics ecosystem relies on continuous training, dataset diversity, and adherence to safety standards. Tour guide robots utilize datasets like RH20T for interactive navigation tasks [34], while Classroom Support Robots benefit from Open X-Embodiment for manipulation and task execution workflows. Safety and Security Robots depend on BridgeData V2 for adapting to complex safety protocols in diverse environments, and Academic Assistance Robots rely on OpenLORIS-Object for accurate object recognition and task performance [50,59]. By addressing sim-to-real transfer challenges such as domain randomization, sensor calibration, and incremental fine-tuning, campus service robots can be rigorously trained in virtual environments before deployment. Simulation tools like Isaac Gym, MuJoCo, and Gazebo offer scalable platforms for prototyping and testing. This iterative cycle of training, validation, and deployment ensures that campus robotics systems remain adaptable, safe, and aligned with institutional goals.

4.1. Popular Training Frameworks and Reliable Simulation Platforms

Reliable simulation platforms such as Isaac Gym, Gazebo, PyBullet, and MuJoCo have become essential for prototyping and validating robotic behaviors before real-world deployment. These platforms provide robust physics engines, customizable environments, and support for GPU-accelerated parallel simulations [60–62]. Each platform has its strengths, and selecting the right one depends on task complexity, resource availability, and specific robotic objectives. As shown in Table 9, each simulation framework—Isaac Gym, MuJoCo, Gazebo, and PyBullet—offers unique strengths and trade-offs in physics accuracy, scalability, and sensor integration, guiding researchers in selecting the appropriate tool based on task complexity and deployment goals.

Table 9. Pros and cons of simulation frameworks.

Framework	Pros	Cons
Isaac Gym	GPU-accelerated training, suitable for reinforcement learning	Resource-intensive
MuJoCo	Precise mechanical and physics modeling	Computationally expensive
Gazebo	Realistic sensor integration	Slower performance for large datasets
PyBullet	Lightweight, fast for prototyping	Limited sensor fidelity

4.2. Benchmark Datasets for Robotic Learning

For campus service robots, RH20T and Open X-Embodiment excel in diverse skill learning and multi-modal integration. BridgeData V2 provides scalability for multi-task scenarios, while OpenLORIS-Object focuses on real-time object recognition under changing environmental conditions. As detailed in Table 10, these benchmark datasets—ranging from RH20T to OpenLORIS-Object—support diverse robotic competencies by offering varied task types, sensor modalities, and generalization capabilities critical for training and evaluating campus service robots.

Table 10. Overview of benchmark datasets.

Dataset	Tasks	Modalities	Focus Area
RH20T	140+ manipulation tasks	Visual, tactile, audio, proprioceptive	Diverse skill learning, one-shot imitation
Open X-Embodiment	527 skills across 22 robot types	Multi-modal (vision, proprioception, language)	Multi-robot generalization
BridgeData V2	13 core skills, 24 environments	Goal- and language-conditioned data	Cross-environment generalization
RoboGen	Infinite tasks via generative models	Multi-modal synthetic data	RL, long-horizon tasks
OpenLORIS-Object	Object recognition tasks	RGB-D, point clouds	Lifelong object recognition

4.3. Bridging Simulation and Reality

The sim-to-real gap refers to the challenge of transferring robotic skills and policies trained in simulation to real-world environments, where differences in sensor noise, physics, friction, lighting, and object textures often hinder seamless deployment. Key approaches to address this gap include the following:

- Domain Randomization: By introducing variability in simulation parameters, including lighting, object positions, and sensor noise, robots are exposed to diverse conditions, increasing their adaptability to real-world uncertainties [63].
- Domain Adaptation: Techniques such as adversarial learning and fine-tuning with real-world data align simulation policies with real-world behavior [37].
- Incremental Fine-Tuning: After initial training in simulation, real-world data is incrementally used to refine robot behavior, addressing simulation-specific biases [40].
- Sensor Calibration and High-Fidelity Modeling: Improved sensor accuracy and high-fidelity physics engines ensure better alignment between simulated and real sensor readings [64].

5. Comparative Results and Benchmark Synthesis

This section synthesizes the performance and impact of the representative systems discussed in Sections 3 and 4, spanning locomotion, semantic navigation, manipulation, voice-based interaction, and supporting datasets. By combining architectural innovation with empirical evidence, we offer a holistic view of how large language models (LLMs) are enabling generalizable, multi-modal, and human-aligned robotic autonomy.

5.1. Locomotion and Loco-Manipulation

LLMs enhance low-level control and long-horizon planning in locomotion tasks. SayTap integrates LLM-prompted binary contact patterns with deep reinforcement learning (DRL) to stabilize gait, increasing task success rates by over 20% in zero-shot trials. WildLMa handles complex outdoor terrain with 93% success across 12 novel conditions. For loco-manipulation, BETR-XP-LLM adapts behavior trees in real-time, improving execution speed by 32% and reducing task failures by 27%, showcasing the synergy between LLM planning and reactive motion controllers.

5.2. Navigation and Semantic Planning

Semantic navigation systems powered by LLMs excel at dynamic, goal-conditioned routing. MapGPT employs topological prompting to achieve 91% success in long-range tasks, while SayNav maintains 85% accuracy in real-time obstacle-rich scenes using on-the-fly 3D scene graphs. TrustNavGPT uniquely integrates speech-affect analysis for ambiguous voice navigation, improving robustness by 18% over static STT-based baselines and achieving 83% success under noisy conditions.

5.3. Physical Interaction and Physics-Informed Manipulation

Manipulation capabilities are significantly strengthened through semantic understanding and physics-informed learning. LLM+A improves grasp success by 17% using affordance-based prompting. 3D-LOTUS++ enhances occluded object interaction accuracy by 12%, and CriticGPT reduces assembly task failures by 24% via runtime correction. Physics-informed neural networks (PINNs), such as PINN-Ray, accurately model the deformation of Fin Ray grippers with 93% precision. When paired with RL, PINNs improve trajectory fidelity by 18% and reduce overshoot by 22%, offering precise, compliant manipulation in contact-rich environments.

5.4. Voice-Based Interaction and Control Interfaces

Voice-based systems act as a cross-domain interface, integrating naturally with locomotion, navigation, and manipulation. TrustNavGPT detects vocal affect (e.g., pitch, speed) to assess user intent, achieving a 5.7% word error rate and 83% navigation reliability in noisy settings. VoicePilot interprets modifier-rich instructions (e.g., "slowly", "after a pause") using GPT-3.5 Turbo, attaining 90% command adherence. Hybrid designs combining voice with mobile fallback improve execution responsiveness by 22%, especially under ambiguity. These results affirm the role of voice as a core input modality in real-world human–robot collaboration.

5.5. Training Datasets and Evaluation Benchmarks

High-quality datasets underpin generalization and real-world applicability. RH20T includes 140+ skill-conditioned tasks with rich multi-modal inputs, enabling one-shot imitation. Open X-Embodiment spans 527 skills across 22 robotic embodiments, supporting policy transfer. BridgeData V2 and OpenLORIS-Object contribute scenario variability, helping assess perception stability and semantic generalization in cluttered or shifting environments.

6. Summary

This paper surveys the integration of large language models (LLMs) into autonomous robotics, focusing on locomotion, voice-based interaction, and robust training frameworks. LLMs have emerged as a powerful tool, enabling robots to interpret high-level natural language commands and translate them into actionable low-level motor controls [1,9,65,66]. By combining LLM-driven reasoning with reinforcement learning (RL) and multi-modal datasets, robots can now execute long-horizon tasks, such as climbing stairs to press a button or autonomously navigating dynamic environments [67,68].

In locomotion, LLMs have been used to enhance contact pattern generation for quadrupedal robots and fine-tune motor control policies [15,25,69]. Techniques like hierarchical reinforcement learning (HRL) and natural language processing (NLP) enable precise task decomposition, ensuring smooth coordination between high-level planning and real-time actuation. Adaptive execution frameworks, such as BETR-XP-LLM, further improve error recovery and robustness in complex terrains.

For voice-based interaction, the integration of speech-to-text (STT) systems and edge-cloud hybrid architectures enhances robots' responsiveness to natural voice commands. Tools like OpenAI Whisper and TrustNavGPT have demonstrated success in noisy environments, enabling context-aware navigation and object interaction through voice prompts. Mobile applications complement these systems, offering fallback mechanisms for robust control in edge cases [70].

In terms of training frameworks and datasets, simulation platforms like Isaac Gym, Gazebo, and MuJoCo provide scalable environments for prototyping and iterative training [71]. Benchmark datasets, such as RH20T, Open X-Embodiment, and BridgeData V2, ensure diverse task learning, adaptive behavior, and multi-modal integration. Techniques like domain randomization, sensor calibration, and incremental fine-tuning help bridge the simulation-to-reality gap, ensuring smooth real-world deployment.

Moving forward, the focus will be on creating more context-aware, adaptable, and human-like robotic systems capable of multi-task execution across campus environments. Enhanced accent adaptation, real-time environment mapping, and adaptive reasoning will play a pivotal role in addressing persistent challenges [72]. By leveraging the synergy between LLM-based semantic reasoning, controller-driven precision, and multi-modal data integration, we aim to realize a new generation of intelligent, multi-functional robots capable of classroom assistance, security patrolling, and interactive campus tours.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L.; software, Y.L.; validation, Y.L. and Q.S.; formal analysis, Y.L.; investigation, Y.L.; resources, Q.S.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L., Q.S. and D.R.K.; visualization, Y.L.; supervision, Q.S.; project administration, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fan, Y.; Pei, Z.; Wang, C.; Li, M.; Tang, Z.; Liu, Q. A Review of Quadruped Robots: Structure, Control, and Autonomous Motion. *Adv. Intell. Syst.* **2024**, *6*, 2300783. [CrossRef]
2. Zeng, F.; Gan, W.; Wang, Y.; Liu, N.; Yu, P.S. Large Language Models for Robotics: A Survey. *arXiv* **2023**, arXiv:2311.07226.
3. Tang, C.; Abbatematteo, B.; Hu, J.; Chandra, R.; Martín-Martín, R.; Stone, P. Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. *arXiv* **2024**, arXiv:2408.03539.
4. Jain, D.; Iscen, A.; Caluwaerts, K. Hierarchical Reinforcement Learning for Quadruped Locomotion. *arXiv* **2019**, arXiv:1905.08926.
5. Firoozi, R.; Tucker, J.; Tian, S.; Majumdar, A.; Sun, J.; Liu, W.; Zhu, Y.; Song, S.; Kapoor, A.; Hausman, K.; et al. Foundation Models in Robotics: Applications, Challenges, and the Future. *Int. J. Robot. Res.* **2025**, *44*, 701–739. [CrossRef]
6. Xu, Z.; Wu, K.; Wen, J.; Li, J.; Liu, N.; Che, Z.; Tang, J. A Survey on Robotics with Foundation Models: Toward Embodied AI. *arXiv* **2024**, arXiv:2402.02385.
7. Han, C.; Lee, J.; Lee, H.; Sim, Y.; Jeon, J.; Jun, M.B.-G. Zero-Shot Autonomous Robot Manipulation via Natural Language. *Manuf. Lett.* **2024**, *42*, 16–20. [CrossRef]
8. Xiao, X.; Liu, J.; Wang, Z.; Zhou, Y.; Qi, Y.; Jiang, S.; He, B.; Cheng, Q. Robot Learning in the Era of Foundation Models: A Survey. *Neurocomputing* **2025**, *638*, 129963. [CrossRef]
9. Ouyang, Y.; Li, J.; Li, Y.; Li, Z.; Yu, C.; Sreenath, K.; Wu, Y. Long-Horizon Locomotion and Manipulation on a Quadrupedal Robot with Large Language Models. *arXiv* **2024**, arXiv:2404.05291.
10. Yang, Y.; Shi, G.; Lin, C.; Meng, X.; Scalise, R.; Castro, M.G.; Yu, W.; Zhang, T.; Zhao, D.; Tan, J.; et al. Agile Continuous Jumping in Discontinuous Terrains. *arXiv* **2024**, arXiv:2409.10923.
11. Jeong, H.; Lee, H.; Kim, C.; Shin, S. A Survey of Robot Intelligence with Large Language Models. *Appl. Sci.* **2024**, *14*, 8868. [CrossRef]
12. Liang, B.; Sun, L.; Zhu, X.; Zhang, B.; Xiong, Z.; Li, C.; Sreenath, K.; Tomizuka, M. Adaptive Energy Regularization for Autonomous Gait Transition and Energy-Efficient Quadruped Locomotion. *arXiv* **2024**, arXiv:2403.20001.
13. Hahn, D.; Banzet, P.; Bern, J.M.; Coros, S. Real2Sim: Visco-Elastic Parameter Estimation from Dynamic Motion. *ACM Trans. Graph.* **2019**, *38*, 1–13. [CrossRef]
14. Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. RT-1: Robotics Transformer for Real-World Control at Scale. *arXiv* **2022**, arXiv:2212.06817.
15. Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. *arXiv* **2023**, arXiv:2307.15818.
16. Hoeller, D.; Wellhausen, L.; Farshidian, F.; Hutter, M. Learning a State Representation and Navigation in Cluttered and Dynamic Environments. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5081–5088. [CrossRef]
17. Qiu, R.-Z.; Song, Y.; Peng, X.; Suryadevara, S.A.; Yang, G.; Liu, M.; Ji, M.; Jia, C.; Yang, R.; Zou, X.; et al. WildLMa: Long Horizon Loco-Manipulation in the Wild. *arXiv* **2024**, arXiv:2411.15131.
18. Fan, Z.; Gao, X.; Mirchev, M.; Roychoudhury, A.; Tan, S.H. Automated Repair of Programs from Large Language Models. In Proceedings of the 45th International Conference on Software Engineering, Melbourne, Australia, 14–20 May 2023; IEEE Press: Piscataway, NJ, USA, 2023; pp. 1469–1481.
19. Tang, Y.; Yu, W.; Tan, J.; Zen, H.; Faust, A.; Harada, T. SayTap: Language to Quadrupedal Locomotion. *arXiv* **2023**, arXiv:2306.07580.
20. Sun, X.; Zhang, Y.; Tang, X.; Bedi, A.S.; Bera, A. TrustNavGPT: Modeling Uncertainty to Improve Trustworthiness of Audio-Guided LLM-Based Robot Navigation. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024.
21. Rajvanshi, A.; Sikka, K.; Lin, X.; Lee, B.; Chiu, H.-P.; Velasquez, A. SayNav: Grounding Large Language Models for Dynamic Planning to Navigation in New Environments. In Proceedings of the Thirty-Fourth International Conference on Automated Planning and Scheduling, Banaff, AB, Canada, 1–6 June 2024; 34, pp. 464–474. [CrossRef]
22. Liang, J.; Xia, F.; Yu, W.; Zeng, A.; Arenas, M.G.; Attarian, M.; Bauza, M.; Bennice, M.; Bewley, A.; Dostmohamed, A.; et al. Learning to Learn Faster from Human Feedback with Language Model Predictive Control. *arXiv* **2024**, arXiv:2402.11450.

23. Chen, J.; Lin, B.; Xu, R.; Chai, Z.; Liang, X.; Wong, K.-Y. MapGPT: Map-Guided Prompting with Adaptive Path Planning for Vision-and-Language Navigation. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 11–16 August 2024; Volume 1: Long Papers. Ku, L.-W., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 9796–9810.
24. Meng, S.; Wang, Y.; Yang, C.-F.; Peng, N.; Chang, K.-W. LLM-A*: Large Language Model Enhanced Incremental Heuristic Search on Path Planning. *arXiv* **2024**, arXiv:2407.02511.
25. Styrud, J.; Iovino, M.; Norrlöf, M.; Björkman, M.; Smith, C. Automatic Behavior Tree Expansion with LLMs for Robotic Manipulation. *arXiv* **2024**, arXiv:2409.13356.
26. Venkatesh, V.L.N.; Min, B.-C. ZeroCAP: Zero-Shot Multi-Robot Context Aware Pattern Formation via Large Language Models. *arXiv* **2024**, arXiv:2404.02318.
27. Gu, J.; Stefani, E.; Wu, Q.; Thomason, J.; Wang, X.E. Vision-and-Language Navigation: A Survey of Tasks, Methods, and Future Directions. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Volume 1: Long Papers. pp. 7606–7623.
28. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
29. Gao, J.; Sarkar, B.; Xia, F.; Xiao, T.; Wu, J.; Ichter, B.; Majumdar, A.; Sadigh, D. Physically Grounded Vision-Language Models for Robotic Manipulation. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 12462–12469.
30. Park, S.-M.; Kim, Y.-G. Visual Language Navigation: A Survey and Open Challenges. *Artif. Intell. Rev.* **2022**, *56*, 365–427. [CrossRef]
31. Liu, M.; Xiao, J.; Li, Z. Deployment of Whole-Body Locomotion and Manipulation Algorithm Based on NMPC Onto Unitree Go2Quadruped Robot. In Proceedings of the 2024 6th International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 21–24 August 2024; pp. 1–6.
32. Atuhurra, J. Leveraging Large Language Models in Human-Robot Interaction: A Critical Analysis of Potential and Pitfalls. *arXiv* **2024**, arXiv:2405.00693.
33. Maranto, D. LLMSat: A Large Language Model-Based Goal-Oriented Agent for Autonomous Space Exploration. *arXiv* **2024**, arXiv:2405.01392.
34. Fang, H.-S.; Fang, H.; Tang, Z.; Liu, J.; Wang, J.; Zhu, H.; Lu, C. RH20T: A Comprehensive Robotic Dataset for Learning Diverse Skills in One-Shot. *arXiv* **2023**, arXiv:2307.00595.
35. Wang, J.; Tsagarakis, N. Grounding Language Models in Autonomous Loco-Manipulation Tasks. *arXiv* **2024**, arXiv:2409.01326.
36. Wang, X.; Gupta, A. Unsupervised Learning of Visual Representations Using Videos. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 2794–2802.
37. Hua, P.; Liu, M.; Macaluso, A.; Lin, Y.; Zhang, W.; Xu, H.; Wang, L. GenSim2: Scaling Robot Data Generation with Multi-Modal and Reasoning LLMs. *arXiv* **2024**, arXiv:2410.03645.
38. Latif, E. 3P-LLM: Probabilistic Path Planning Using Large Language Model for Autonomous Robot Navigation. *arXiv* **2024**, arXiv:2403.18778.
39. Doma, P.; Arab, A.; Xiao, X. LLM-Enhanced Path Planning: Safe and Efficient Autonomous Navigation with Instructional Inputs. *arXiv* **2024**, arXiv:2412.02655.
40. Ahmidi, N.; Tao, L.; Sefati, S.; Gao, Y.; Lea, C.; Haro, B.B.; Zappella, L.; Khudanpur, S.; Vidal, R.; Hager, G.D. A Dataset and Benchmarks for Segmentation and Recognition of Gestures in Robotic Surgery. *IEEE Trans. Biomed. Eng.* **2017**, *64*, 2025–2041. [CrossRef]
41. Kong, X.; Zhang, W.; Hong, J.; Braunl, T. Embodied AI in Mobile Robots: Coverage Path Planning with Large Language Models. *arXiv* **2024**, arXiv:2407.02220.
42. Badr, A.; Abdul-Hassan, A. A Review on Voice-Based Interface for Human-Robot Interaction. *Iraqi J. Electr. Electron. Eng.* **2020**, *16*, 91–102. [CrossRef]
43. Padmanabha, A.; Yuan, J.; Gupta, J.; Karachiwalla, Z.; Majidi, C.; Admoni, H.; Erickson, Z. VoicePilot: Harnessing LLMs as Speech Interfaces for Physically Assistive Robots. In Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology, Pittsburgh, PA, USA, 13–16 October 2024; Association for Computing Machinery: New York, NY, USA, 2024; pp. 1–18.
44. Liu, J.; Borja, P.; Santina, C.D. Physics-Informed Neural Networks to Model and Control Robots: A Theoretical and Experimental Investigation. *arXiv* **2023**, arXiv:2305.05375. [CrossRef]
45. PINN-Ray: A Physics-Informed Neural Network to Model Soft Robotic Fin Ray Fingers. Available online: https://arxiv.org/html/2407.08222v1?utm_source=chatgpt.com (accessed on 30 May 2025).

46. Liu, Y.; Bao, Y.; Cheng, P.; Shen, D.; Chen, G.; Xu, H. Enhanced Robot State Estimation Using Physics-Informed Neural Networks and Multimodal Proprioceptive Data. In Proceedings of the Sensors and Systems for Space Applications XVII, National Harbor, MD, USA, 21–26 April 2024; SPIE: New York, NY, USA, 2024; Volume 13062, pp. 144–160.
47. Ni, R.; Qureshi, A.H. Physics-Informed Neural Networks for Robot Motion under Constraints. In Proceedings of the RoboNerF: 1st Workshop on Neural Fields in Robotics at ICRA, Yokohama, Japan, 13 May 2024.
48. Deuerlein, C.; Langer, M.; Seßner, J.; Heß, P.; Franke, J. Human-Robot-Interaction Using Cloud-Based Speech Recognition Systems. *Procedia CIRP* **2021**, *97*, 130–135. [CrossRef]
49. She, Q.; Feng, F.; Hao, X.; Yang, Q.; Lan, C.; Lomonaco, V.; Shi, X.; Wang, Z.; Guo, Y.; Zhang, Y.; et al. OpenLORIS-Object: A Robotic Vision Dataset and Benchmark for Lifelong Deep Learning. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 4767–4773.
50. Walke, H.R.; Black, K.; Zhao, T.Z.; Vuong, Q.; Zheng, C.; Hansen-Estruch, P.; He, A.W.; Myers, V.; Kim, M.J.; Du, M.; et al. BridgeData V2: A Dataset for Robot Learning at Scale. In Proceedings of the 7th Conference on Robot Learning, Atlanta, GA, USA, 6–9 November 2023; pp. 1723–1736.
51. Basyal, L. Voice Recognition Robot with Real-Time Surveillance and Automation. *arXiv* **2023**, arXiv:2312.04072.
52. Rodriguez-Losada, D.; Matia, F.; Galan, R.; Hernando, M.; Manuel, J.; Manuel, J. Urbano, an Interactive Mobile Tour-Guide Robot. In *Advances in Service Robotics*; Seok, H., Ed.; InTech: Vienna, Austria, 2008; ISBN 978-953-7619-02-2.
53. Gupta, A.; Murali, A.; Gandhi, D.P.; Pinto, L. Robot Learning in Homes: Improving Generalization and Reducing Dataset Bias. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Curran Associates, Inc.: Nice, France, 2018; Volume 31.
54. Ren, P.; Li, M.; Luo, Z.; Song, X.; Chen, Z.; Liufu, W.; Yang, Y.; Zheng, H.; Xu, R.; Huang, Z.; et al. InfiniteWorld: A Unified Scalable Simulation Framework for General Visual-Language Robot Interaction. *arXiv* **2024**, arXiv:2412.05789.
55. Wang, Y.; Xian, Z.; Chen, F.; Wang, T.-H.; Wang, Y.; Fragkiadaki, K.; Erickson, Z.; Held, D.; Gan, C. RoboGen: Towards Unleashing Infinite Data for Automated Robot Learning via Generative Simulation. *arXiv* **2024**, arXiv:2311.01455.
56. Diallo, A.D.; Gobee, S.; Durairajah, V. Autonomous Tour Guide Robot Using Embedded System Control. *Procedia Comput. Sci.* **2015**, *76*, 126–133. [CrossRef]
57. Chen, F.; Xu, B.; Hua, P.; Duan, P.; Yang, Y.; Ma, Y.; Xu, H. On the Evaluation of Generative Robotic Simulations. *arXiv* **2024**, arXiv:2410.08172.
58. Chen, S.; Wan, Z.; Yan, S.; Zhang, C.; Zhang, W.; Li, Q.; Zhang, D.; Farrukh, F.U.D. SLR: Learning Quadruped Locomotion without Privileged Information. *arXiv* **2024**, arXiv:2406.04835.
59. Collaboration, O.X.-E.; O'Neill, A.; Rehman, A.; Gupta, A.; Maddukuri, A.; Gupta, A.; Padalkar, A.; Lee, A.; Pooley, A.; Gupta, A.; et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. *arXiv* **2024**, arXiv:2310.08864.
60. Cherner, Y.; Lotring, A.; Campbell, T.; Klein, R. Innovative Simulation Based Online System for Learning Engineering and Training Sailors' Technical Skills. In Proceedings of the 2006 Annual Conference & Exposition, Chicago, IL, USA, 18–21 June 2006.
61. Katara, P.; Xian, Z.; Fragkiadaki, K. Gen2Sim: Scaling up Robot Learning in Simulation with Generative Models. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 6672–6679.
62. Sobanbabu, N.; He, G.; He, T.; Yang, Y.; Shi, G. Sampling-Based System Identification with Active Exploration for Legged Robot Sim2Real Learning. *arXiv* **2025**, arXiv:2505.14266.
63. Ebert, F.; Yang, Y.; Schmeckpeper, K.; Bucher, B.; Georgakis, G.; Daniilidis, K.; Finn, C.; Levine, S. Bridge Data: Boosting Generalization of Robotic Skills with Cross-Domain Datasets. *arXiv* **2021**, arXiv:2109.13396.
64. Dentler, J.; Kannan, S.; Bezzaoucha, S.; Olivares-Mendez, M.A.; Voos, H. Model Predictive Cooperative Localization Control of Multiple UAVs Using Potential Function Sensor Constraints. *Auton. Robot.* **2019**, *43*, 153–178. [CrossRef]
65. Dianatfar, M.; Latokartano, J.; Lanz, M. Review on Existing VR/AR Solutions in Human–Robot Collaboration. *Procedia CIRP* **2021**, *97*, 407–411. [CrossRef]
66. Kaczmarek, W.; Panasiuk, J.; Borys, S.; Banach, P. Industrial Robot Control by Means of Gestures and Voice Commands in Off-Line and On-Line Mode. *Sensors* **2020**, *20*, 6358. [CrossRef]
67. Joseph, P.; Plozza, D.; Pascarella, L.; Magno, M. Gaze-Guided Semi-Autonomous Quadruped Robot for Enhanced Assisted Living. In Proceedings of the 2024 IEEE Sensors Applications Symposium (SAS), Naples, Italy, 23–25 July 2024; pp. 1–6.
68. Sela, M.; Xu, P.; He, J.; Navalpakkam, V.; Lagun, D. GazeGAN—Unpaired Adversarial Image Generation for Gaze Estimation. *arXiv* **2017**, arXiv:1711.09767.
69. Cheng, A.-C.; Ji, Y.; Yang, Z.; Gongye, Z.; Zou, X.; Kautz, J.; Bıyık, E.; Yin, H.; Liu, S.; Wang, X. NaVILA: Legged Robot Vision-Language-Action Model for Navigation. *arXiv* **2025**, arXiv:2412.04453.
70. Fu, Z.; Zhao, T.Z.; Finn, C. Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation. *arXiv* **2024**, arXiv:2401.02117.

71. Chatzilygeroudis, K.; Fichera, B.; Lauzana, I.; Bu, F.; Yao, K.; Khadivar, F.; Billard, A. Benchmark for Bimanual Robotic Manipulation of Semi-Deformable Objects. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2443–2450. [CrossRef]
72. Vijayakumar, S.; D'souza, A.; Shibata, T.; Conradt, J.; Schaal, S. Statistical Learning for Humanoid Robots. *Auton. Robot.* **2002**, *12*, 55–69. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.