Review | Received 27 November 2025; Revised 12 February 2026; Accept 10 April 2026; Published 27 April 2026
https://doi.org/10.55092/rl20260010

The evolution of end-to-end Vision-Language-Action (VLA) architectures in robotics

Jingjing Pei[1,†], Xiaoyin Zheng[2,†], Yang Liu[3], Bike Zhu[1], Daifeng Wang[1], Jiajun An[1], Richard Voyles[4] and Xin Ma[1,*]

[1] Department of Mechanical and Energy Engineering, Institute for Robotics Research, Southern University of Science and Technology, Shenzhen 518055, China
[2] The Autonomous Driving Center, XMotors.ai., Inc., Santa Clara 95054, USA
[3] State Key Laboratory of High-performance Precision Manufacturing, Dalian University of Technology, Dalian 116024, China
[4] Department of Computer Science and Engineering, The University of Texas at Arlington, Arlington 76019, USA

† These authors contributed equally to this work.

* Correspondence author; E-mail: max6@sustech.edu.cn.

Highlights:

- Decomposition of VLA architectures into perception, fusion, and action modules.
- Clear distinction between end-to-end and hierarchical architectures.
- Trade-off analysis of latency and precision for architecture selection.

Abstract: Vision-Language-Action (VLA) models represent a fundamental architectural shift in robotic learning, replacing modular perception-reasoning-control pipelines with unified frameworks that jointly optimize multimodal understanding and motor control. While large language models (LLMs) have enabled natural language grounding in robotics, the core challenge remains how to effectively fuse visual perception, linguistic reasoning, and continuous action generation within a single coherent architecture. This survey provides a systematic decomposition of modern VLA systems into three critical components, including multimodal perception encoders, cross-modal fusion mechanisms, and action decoders. We also critically evaluate the impact of design choices on generalization, sample efficiency, and task complexity. We distinguish two dominant architectural paradigms: end-to-end models that directly map observations to actions through learned representations, and hierarchical models that decompose tasks into explicit planning and execution stages. Through comparative analysis of their trade-offs in zero-shot generalization, interpretability, and long-horizon performance, we identify key open challenges in semantic grounding, spatial reasoning, and sim-to-real transfer that will determine the viability of VLAs for real-world deployment.

Keywords: Vision-Language-Action (VLA); embodied AI; robot learning

1. Introduction

Over the past few years, robotic learning has undergone a paradigm shift, from modular and handcrafted pipelines toward fully integrated, data-driven, and semantically grounded architectures. Traditional robotic systems were built upon a serial perception-planning-control pipeline, where each component was engineered independently: visual modules extracted object-level information, planning modules optimized geometric trajectories, and control modules executed low-level actuation through proportional integral derivative (PID) or model predictive control (MPC). While such systems achieved high task precision and interpretability, they lacked generality. Each new task demanded task-specific feature design and interface tuning, preventing transferability and semantic consistency across domains.

The emergence of Transformer-based architectures and the availability of large-scale demonstration datasets have enabled a new generation of end-to-end differentiable systems, which directly map sensory inputs to motor outputs through unified neural representations. Models such as a low-cost open-source hardware system for bimanual teleoperation (ALOHA) [1] used dual-arm imitation learning with a Transformer to map visual inputs to actions, enabling end-to-end imitation from human demonstration. It showed strong generalization in physical manipulation but lacked language instructions. Gato [2] introduced a unified token sequence approach, processing images, text, and actions via a single autoregressive Transformer. It demonstrated multi-task, cross-modal learning but used language only for description, not reasoning. And Robotics Transformer 1 (RT-1) [3] trained on 130k real-world demonstrations to directly map vision to actions. It incorporated language for task conditioning but remained weakly language-supervised, without deep semantic reasoning.

However, despite their strong generalization across physical skills, early end-to-end systems remained semantically shallow. They primarily learned reactive mappings between pixels and motor commands, devoid of reasoning, abstraction, or understanding of human intent. The system could “see and act,” but not “understand and plan.” To move toward general embodied intelligence, a new cognitive dimension had to be integrated: language.

The advent of large language models (LLMs) [4,5] brought precisely this missing link: a bridge from differentiable control to semantic reasoning. By introducing linguistic grounding into end-to-end robotic frameworks, researchers transitioned from purely reactive perception-action coupling to Vision-Language-Action (VLA) [6] paradigms capable of comprehension, reasoning, and multi-step planning. LLMs act as a semantic hub, mapping perceptual representations into structured language spaces and generating executable action goals through reasoning. This enables robots to not only execute learned behaviors but also interpret novel, compositional instructions such as “place the red cup to the right of the blue bowl” [7].

Early explorations like SayCan (Say: large language model, Can: Affordance Functions) [8] combined LLMs with value functions, enabling high-level semantic planning constrained by physical feasibility. Pathways language model with embodied capabilities (PaLM-E) [9] extended this principle by embedding a LLM directly within a multimodal Transformer, enabling open-vocabulary grounding and cross-task transfer. And Robotics Transformer 2 (RT-2) [10] pushed further by aligning visual-language pretraining (VLP) models with control policies, achieving semantic-to-action generalization.

Subsequent research has systematically advanced the fusion and reasoning for robotics. VisuoMotor Attention agent (VIMA) [11] pioneered this by using multi-modal prompts to output decisions and actions. Open vision-language-action (OpenVLA) [12] accelerated the paradigm with an open-source platform, enabling robust skill transfer across diverse hardware. Further abstracting the stack, a unified Vision-Language-Action model series (VLA-OS) [13] conceptualized the robot as an operating system, where a central LLM orchestrates specialized models, enabling a new hierarchy of compositional reasoning and task decomposition with a hierarchical VLA.

This evolutionary trajectory from modular control to semantically orchestrated systems is visually synthesized in Figure 1. The figure adopts a four-stage chronological and architectural taxonomy to illustrate the paradigm shifts in VLA design: (1) Pre-VLA Era: Characterized by decoupled perception, planning, and control modules (e.g., PID/MPC controllers), operating without linguistic grounding. (2) Early End-to-End Models: Integrated vision and language inputs into a single Transformer backbone, mapping instructions like “Pick apple” directly to low-level actions, yet lacking semantic reasoning. (3) Introduction of LLMs: Incorporated pre-trained vision-language models (VLMs) as high-level planners, enabling semantic interpretation of instructions (e.g., “Pick dustpan”) and generating structured sub-goals, though still coupled with traditional control interfaces. (4) Hierarchical VLA Architectures: Fully realized hierarchical systems where a central LLM orchestrates multi-modal inputs (vision + language) and decomposes tasks into executable sub-goals through specialized reasoning modules, exemplified by operating system metaphors like VLA-OS. This structured visualization not only chronicles the historical progression but also highlights the increasing role of language as a unifying interface for perception, reasoning, and action in embodied AI.

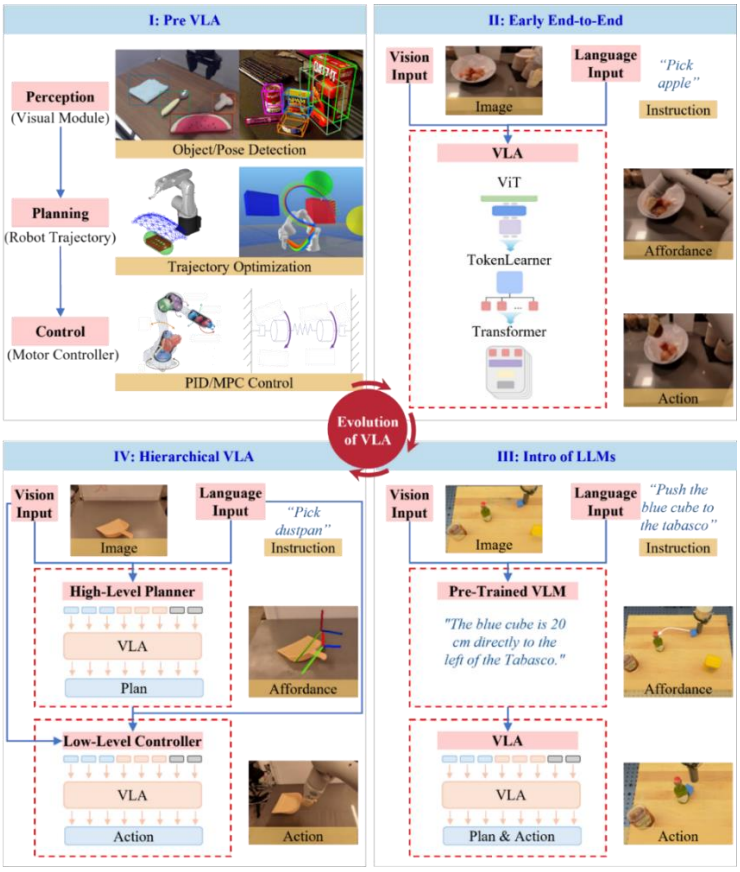

Figure 1. A conceptual taxonomy of VLA architectures.

Collectively, these developments represent a cognitive upgrade of the end-to-end paradigm. The incorporation of LLMs transforms robotic learning from reactive control toward semantic decision-making, which is a shift from differentiability to deliberation [14]. Robots can now interpret human intent, decompose tasks, and reason about cause and effect, moving beyond simple mimicry. This evolution has given rise to the modern VLA system, a unified framework that integrates perception, reasoning, and control into a single, learnable pipeline.

To provide a structured understanding of modern VLA systems, this review is organized into several main parts, which is illustrated in Figure 2. In Section 2, we dissect the core components of a VLA architecture: the multimodal perception encoder, which extracts and aligns visual, linguistic, and spatial features; the fusion and reasoning module, which is responsible for transforming multimodal sensing features into executable task plans; and the action decoder, which translates semantic plans into executable control commands. Each component is analyzed in terms of its architectural evolution, key techniques, and representative models. Building on this component-level analysis, Section 3 contrasts the two prevailing architectural paradigms: the end-to-end integrated approach, which processes perception and action within a single model, and the hierarchical approach, which decouples high-level planning from low-level execution. We examine the design principles, advantages, and limitations of each paradigm, and highlight recent innovations that push the boundaries of semantic generalization, efficiency, and long-horizon robustness. In Section 4, we synthesize the core challenges confronting current VLA architectures and explores promising avenues for future research.

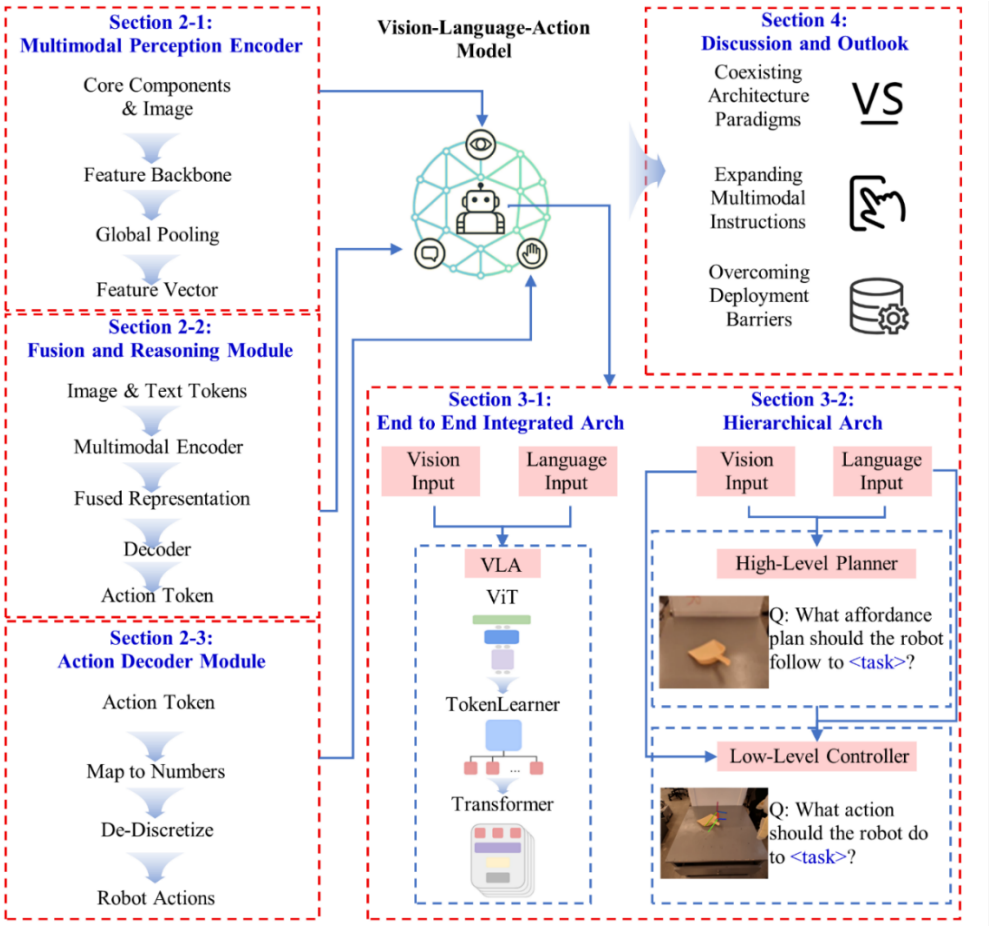

Figure 2. Structure of this survey.

2. The components of the end-to-end VLA system

Modern VLA architectures typically consist of three interconnected modules: multimodal perception encoder, fusion and reasoning module, and action decoder. Together, these modules realize an end-to-end differentiable and semantically grounded control pipeline, allowing robots to perceive, reason, and act through a unified learning process.

2.1. Multimodal perception encoder

The multimodal perception encoder serves as the sensory front-end of VLA systems. Its core objective is to extract high-quality, expressive unimodal features from raw, heterogeneous data. These features form the foundation for downstream fusion and reasoning, with their quality and dimensionality directly determining the system's breadth and depth in perceiving the world. The evolution of this module is primarily characterized by the continuous enrichment and expansion of visual features.

General-purpose visual backbones have become the standard for 2D semantic extraction. Transformer-based architectures, such as Vision Transformer (ViT) [15–18] and ConvNeXt [19,20], have emerged as the de facto solutions for extracting semantic features from RGB images due to their powerful capabilities in modeling long-range dependencies and capturing contextual semantics [21].

Expanding perception from 2D semantics to 3D geometry and physical attributes [22,23]. To overcome the limitations of pure RGB vision in depth perception and spatial understanding, recent research focus has shifted towards 3D scene understanding. Representative works like Uni3D [24] and EmbodiedGPT [25] incorporate modalities such as depth maps, point clouds, and tactile feedback, relying on 3D perception backbones like PointNet++ [26] and BEVFormer [27] specifically designed to extract spatial geometric features and physical relationships between objects. This evolution lays the perceptual groundwork for spatial reasoning in dynamic and cluttered environments for robots.

Task-specific specialized encoding architectures. Furthermore, to achieve peak performance in specific tasks, a range of specialized encoders has emerged. For instance, OpenFrontier [28] is designed for open-world navigation, extracting features for frontier exploration; DepthVLA [29] integrates depth prediction modules to enhance 3D perception, while NICE [30] improves robustness through programmatic scene editing. And TacRefineNet [31] demonstrates the potential of pure tactile sensing for extracting fine-grained physical contact features in millimeter-precision manipulation tasks.

The evolutionary path of perception encoders clearly manifests as a progression from general 2D semantic feature extraction, to specialized 3D geometric and physical feature extraction, and further to task-oriented, highly optimized feature design. This process enables VLA systems to "see" and "feel" an increasingly diverse range of environmental information.

2.2. Fusion and reasoning module

The fusion and reasoning module is the cognitive core of VLA systems, responsible for transforming multimodal perceptual features into executable task plans. This module centers around LLMs, and its functionality can be deconstructed into two closely linked levels: semantic fusion and task reasoning and planning. Semantic fusion aims to construct a unified cross-modal semantic representation, serving as

the foundation for reasoning [32]; task reasoning and planning then performs logical decomposition and policy generation based on this representation [33].

2.2.1. Semantic fusion and cross-modal alignment

The core of semantic fusion lies in achieving deep cross-modal alignment. Its technical evolution clearly demonstrates a paradigm shift from "implicit task-driven alignment" to "explicit general-purpose semantic alignment".

Early frameworks learned cross-modal associations implicitly through end-to-end training on robotic control data. In such models, visual and language encoders were typically pre-trained independently, and their alignment relied on downstream fusion modules passively learning under the supervision of action sequences. For example, RT-1 [3] used EfficientNet and a T5-style text encoder to process images and instructions respectively, learning to associate language instructions with visual scenes to generate actions via a FiLM-Transformer fusion module. VIMA [11] introduced a multimodal transformer encoder, fusing instructions and visual inputs via attention mechanisms within the task context. While this approach enabled efficient command parsing and in-context learning, the alignment learned on limited task data resulted in shallow semantic representations, limiting complex reasoning capabilities.

The introduction of dedicated vision-language alignment models elevated cross-modal alignment to an explicit, general-purpose pre-training capability. Models like SigLIP [34] from Google, pre-trained on massive image-text pairs, have the core objective of directly pulling the representations of related visual and linguistic concepts closer in a shared embedding space using techniques like contrastive learning. They provide VLA systems with a high-quality, highly robust alignment foundation, significantly improving generalization in robotic tasks [35,36]. Energy-based control methods provide a theoretical framework for this alignment, achieving stable matching of cross-modal semantics by minimizing an energy function [37].

Currently, large-scale vision-language foundation models act as fully-fledged fusion engines. Models such as InternVL [38], PaliGemma [39], and Qwen2.5-VL [40] are themselves general-purpose VL alignment systems pre-trained on vast web-scale data. Within VLA systems, they are often used as frozen perception backbones, directly providing deeply aligned multimodal context for subsequent reasoning. The success of systems like RT-2 [10] and VLA-OS [13] fully validates the exceptional visual grounding and robustness of such models under real-world conditions. While these large-scale VLMs provide a robust alignment foundation, their effective use in robotics often requires subsequent specialization, such as task-specific fine-tuning or in-context learning, to bridge the gap between web-scale semantics and embodied control.

At the fusion architecture level, token concatenation is a mainstream technique [41–43], where feature sequences from different modalities are concatenated and fed into the LLM backbone, which learns cross-modal relations through its attention mechanism. Cross-attention mechanisms [44–48] enable dynamic, fine-grained cross-modal interaction [49]. Furthermore, techniques like contrastive pre-training [50] and masked modeling [51] are used to enhance alignment quality. Some systems [52,53] also employ unified encoders for deeper fusion.

The research frontier is focused on expanding the dimensions of fusion and improving its efficiency [54,55]. MLA [56] achieves richer sensor fusion by performing token-level contrastive alignment to unify 2D images, 3D point clouds, and tactile signals into a single semantic space. To

enhance efficiency, ContextVLA [57] introduces temporal context compression, embedding multiple frames into compact context tokens for efficient reasoning. HyperVLA [58] explores hypernetwork-based policy generation, achieving significant parameter reduction. In practical applications, systems like Xiaomi EV-AD VLA [59] enhance semantic grounding via text retrieval, while VLA ²[60] expands the system's semantic vocabulary through web-scale retrieval.

2.2.2. Task reasoning and planning

After obtaining the fused semantic representation, the LLM further demonstrates high-level cognitive functions. In task reasoning, LLMs leverage their world knowledge and logical capabilities acquired during pre-training to perform hierarchical reasoning on complex instructions, generating interpretable reasoning chains. The SayCan [8] framework is a hallmark of this capability, utilizing the LLM to decompose abstract user instructions into a sequence of structured, high-level semantic steps, marking VLA's transition from mere action imitation to explicit cognitive planning.

In policy planning, the LLM generates concrete action blueprints based on the reasoning outcome [61], primarily following two paradigms. In end-to-end policies, the LLM directly outputs low-level action commands or parameters, forming a continuous, differentiable mapping from semantic understanding to low-level control [62]. The PaLM-E [9] framework, for instance, inserts visual tokens directly into the language model's input sequence and outputs actions. In hierarchical planning systems, the LLM generates high-level sub-goals, which are then executed by dedicated low-level controllers [63,64]. The VLA-OS [13] model, by introducing multiple planning heads within the backbone network, enables the LLM to explicitly generate intermediate sub-goals in a shared representation space, serving as a typical example of this paradigm.

Current research enhances this module by introducing advanced reasoning frameworks and improving system robustness. In advanced reasoning, VLA-Reasoner [65] significantly enhances decomposition and planning capabilities for long-horizon tasks by integrating Monte Carlo Tree Search; SITCOM [66] employs inference-time trajectory simulation and reward guidance for decision-making; PhysiAgent [67] introduces a modular scaffold comprising planner, monitor, and reflector components, enabling reflective self-correction. In robustness and introspection, INSIGHT [68] introduces metacognitive capabilities through uncertainty detection, allowing the system to recognize its cognitive boundaries; FailSafe [69] designs an integrated failure self-recovery mechanism; and SEAL [70] enhances decision safety through simulation-based candidate action verification. Concurrently, studies like LIBERO-PRO [71] reveal the fragility of current models in complex language understanding. To improve adaptability, MoS-VLA [72] achieves one-shot skill adaptation through a structured skill space; VLM2VLA [73] focuses on language-behavior bridging to prevent catastrophic forgetting during fine-tuning.

The evolution of the fusion and reasoning module reflects the progression of VLA systems from simple perception-action mapping toward "embodied brains" possessing deep semantic understanding, logical reasoning, and autonomous planning capabilities. Centered around LLMs and combined with increasingly sophisticated cross-modal alignment and fusion techniques, this module is driving robotics toward higher levels of autonomous behavior in complex, unstructured environments.

2.3. Action decoder module

Action decoder module generates executable control outputs from the high-level semantic plan. Its primary function is to translate high-level semantic representations or plans produced by the fusion and reasoning module into executable low-level control commands. This transformation enables the system to close the perception-cognition-action loop, forming a unified end-to-end control pipeline in which abstract goals are continuously converted into concrete motor actions. Depending on the representation and generation mechanisms of actions, existing VLA systems employ three principal decoding paradigms: autoregressive sequence generation, diffusion-based policy generation, and flow-matching or fast policy generation. Each method provides a distinct balance between temporal coherence, real-time performance, and representational flexibility in continuous control.

Autoregressive sequence generation represents the earliest and most widely adopted decoding paradigm in VLA systems such as Gato, the RT series, and OpenVLA [12,74]. In this approach, the model encodes visual and linguistic features into a contextual sequence and then predicts the next action token step by step in an autoregressive manner. This design benefits from the strong sequential modeling capability of Transformer architectures and naturally supports imitation learning frameworks such as behavior cloning. The autoregressive formulation provides inherent closed-loop consistency, as each prediction depends on the previously generated outputs, ensuring temporal continuity in long sequences [75,76]. However, discretizing continuous control signals into action tokens leads to a loss of fine-grained continuity, while error accumulation across long sequences may degrade performance over time. Furthermore, sequential token prediction introduces significant inference latency, which limits the applicability of autoregressive decoders in real-time robotic control. Although this formulation ensures temporal coherence, its sequential nature makes it inefficient for fast, continuous control tasks. Several systems mitigate this issue by introducing sliding context windows and sequence truncation to reduce computational delays while maintaining temporal consistency [77,78].

To address some of these limitations, diffusion-based models have emerged as a complementary alternative, offering parallel trajectory generation and reduced autoregressive error accumulation. This approach models the target action trajectory as a continuous signal corrupted by Gaussian noise and reconstructs the desired sequence through an iterative denoising process [79]. Representative studies include Diffusion Policy [80] and Diffusion-VLA [81], which demonstrate the effectiveness of diffusion-based control for complex, high-dimensional robotic actions. Unlike autoregressive models that generate one token at a time, diffusion-based policies can produce entire continuous trajectories in parallel, allowing smooth, temporally coherent control sequences to be generated in a single inference process [82–84]. This parallelism reduces error accumulation and enhances the diversity of generated actions, as diffusion models inherently learn multimodal distributions of feasible trajectories. They also exhibit stable training behavior and are less prone to gradient explosion compared with autoregressive architectures. Recent extensions like Reflect Drive [85] integrate a gradient-free reflection mechanism that generates multiple trajectory proposals and iteratively corrects unsafe waypoints during inference, achieving enhanced safety while maintaining coherent control in autonomous driving domains.

However, diffusion models introduce their own trade-offs, such as higher per-step computational costs due to iterative denoising, potential training instability in high-dimensional action spaces, and less

inherent temporal coherence over very long horizons compared to autoregressive models. These factors must be weighed when selecting a decoding strategy for real-time robotic systems.

In recent years, flow-matching policy generation has gained attention as an efficient deterministic alternative to diffusion models [86,87]. Flow matching reformulates the stochastic denoising process as a deterministic ordinary differential equation that maps noisy samples directly to the target trajectory in a single forward pass [88,89]. This approach significantly reduces inference time while preserving the representational richness of diffusion-based methods. Prominent frameworks, such as π_0 [90], have adopted this design to realize continuous, low-latency control across diverse robotic tasks, including manipulation and navigation.

Beyond these mainstream paradigms, several hybrid or specialized approaches have been proposed to enhance adaptability and robustness in action decoding. The mixture-of-experts architecture, exemplified by models such as MoLE-VLA [91] and AdaMoE [92], employs multiple specialized sub-policies that generate candidate actions in parallel, while a gating or routing module selects the most suitable output based on the current context. MoS-VLA [72] contributes to decoding by generating skill-conditioned policy vectors in a structured skill space, improving adaptability to novel action distributions through basis policy functions and lightweight convex optimization. Imagination Policy [93] proposes an affordance-based representation called Chain of Moving Oriented Keypoints (CoMOK), modeling manipulation as keypoint trajectories to bridge semantic reasoning with fine-grained motion control. Efficiency optimizations include Action-aware Dynamic Pruning (ADP) [94], which introduces training-free dynamic token pruning that adjusts computational load based on motion phase. MG-Select [95] enhances decision quality through multi-generation test-time scaling, sampling several candidate actions and selecting the most confident using confidence-based scores. Compositional methods like General Policy Composition (GPC) [96] enable zero-shot skill combination through mathematical composition of pre-trained policies. For system robustness, A2C2 [97] maintains closed-loop responsiveness through asynchronous correction heads that refine predicted actions during inference. Memory-augmented decoding in Dejavu [98] enables self-improvement through experience retrieval and corrective residuals. The self-correcting controllers in SC-VLA [99] and FailSafe [69] provide adaptive recovery mechanisms that generate and execute corrective actions following detected failures. Additionally, energy-based control methods formulate motion generation as an optimization problem that minimizes an energy function, proving particularly effective for high-dimensional continuous control scenarios [37].

Together, these decoding mechanisms form the output layer of the VLA architecture, bridging the gap between symbolic reasoning and physical embodiment. By continuously converting semantic understanding into motor control, the action decoder completes the closed-loop cycle of perception, reasoning, and execution. The ongoing integration of autoregressive, diffusion-based, and flow-matching policies is progressively enabling VLA systems to achieve both real-time responsiveness and semantic generalization, marking a critical step toward unified, embodied intelligence capable of perceiving, reasoning, and acting in complex, unstructured environments.

This section describe the components of the modern VLA system. the system’s operation begins with the Multimodal Perception Encoder, which has evolved from extracting general 2D semantics to capturing rich 3D geometric, physical, and task-specific features, forming a comprehensive sensory foundation. The Fusion and Reasoning Module, centered on LLMs, then processes this information. It achieves deep cross-modal semantic alignment and performs complex task decomposition and planning,

effectively serving as the system's cognitive core. Finally, the Action Decoder translates high-level plans into executable low-level control commands through various paradigms, including autoregressive, diffusion-based, and flow-matching generation, thereby closing the perception-action loop. Collectively, the synergistic integration of these modules enables the realization of an end-to-end, semantically grounded control pipeline, empowering robots to perceive, reason, and act autonomously in complex, open-world environments. This unified framework marks a significant leap toward genuine embodied intelligence.

2.4. Component performance and analysis

To move beyond a descriptive catalog, this section extracts actionable design principles from the performance data of individual components. The following analysis grounds the components' ratings in empirical evidence, clarifying the fundamental trade-offs that dictate architectural selection for different robotic applications.

Table 1 summarizes the functional components, representative models, and their comparative attributes across three critical dimensions: real-time capability, training/data efficiency, and model complexity. The assessments presented in the table are grounded in empirical metrics and design specifications reported in the associated literature. Below, we provide specific justifications for the ratings of several key component categories, linking the qualitative scales to quantitative evidence from the cited works.

For the Multimodal Perception Encoder Module, the ratings reflect the trade-offs between representation power and practical deployment constraints. Visual encoders such as ViT and ConvNeXt achieve inference speeds suitable for many robotic perception pipelines; for instance, a ViT-B/16 model can process images in tens of milliseconds on modern GPUs [15]. However, their quadratic attention complexity can become a bottleneck for high-resolution or multi-view inputs, preventing them from achieving the very highest control frequencies, hence the "★★☆" real-time rating [17]. These models necessitate large-scale pre-training datasets, such as ImageNet-21K with approximately 14 million images [15] or web-scale image-text pairs for models like SigLIP [34], justifying the "High" data requirement. In terms of model complexity, standard backbones like ViT-Base contain around 86 million parameters, with larger variants scaling to hundreds of millions, placing them in the "High" complexity category [18]. In contrast, using a large multimodal LLM (e.g., PaLM-E, RT-2) as an encoder introduces significantly greater computational demands. Models like PaLM-E, with 562 billion parameters [9], exhibit substantial inference latency, often exceeding hundreds of milliseconds, which limits real-time control ("★☆" rating) and requires distillation techniques for practical use. Their training is predicated on "Extremely High" data requirements, involving trillion-token-scale multimodal corpora [9,10], and they exhibit "Very High" model complexity due to their dense transformer architectures with parameter counts in the billions to hundreds of billions [9,10,13].

Within the Fusion and Reasoning Module, different architectural choices lead to distinct performance profiles. Cross-modal pretraining methods, such as those employing contrastive or masked modeling objectives, provide efficient single-forward-pass embeddings. Models like CLIP (contrastive language-image pre-training) offer relatively fast alignment ("★★☆" real-time), though their integration into a full VLA pipeline adds overhead. Their "High" data requirement stems from the need for massive, curated datasets of paired image-text data, such as LAION-5B with 5.85 billion pairs, for effective

contrastive learning [50,51]. Conversely, hierarchical or planning-head LLM architectures (e.g., VLA-OS, Groot N1) involve more deliberate reasoning processes. The planning stage often requires multiple forward passes for subgoal generation and refinement, introducing significant computational overhead and resulting in a lower "★" real-time capability [63,64]. Training these systems demands "Very High" data investment, specifically large-scale, structured datasets that contain long-horizon task decompositions paired with corresponding low-level actions [63,64].

Table 1. A comparative analysis of core modules in modern VLA systems.

Module	Representative Algorithms/ Models	Core Architecture	Real-time Capability	Training/Data Requirement	Model Complexity	Representative Works
I. Multimodal Perception Encoder Module	Visual Encoders (ViT, ConvNeXt, SigLIP)	Transformer or CNN-Transformer visual backbone	★★☆	High: requires large-scale image-text paired datasets	High	[15–18,34]
	Depth/Multi-Sensor Encoder (Uni3D, EmbodiedGPT)	3D multimodal encoder fusing RGB, depth, and point cloud	★☆	High: multi-sensor fusion datasets	Very High	[24–27]
	Language Encoders (T5, BERT, GPT-family)	Lightweight encoder-decoder or decoder-only text model	★★★	Medium-High: text + command corpus	Medium	[3,11]
	LLM Backbone as Encoder (PaLM, PaLM-E, RT-2)	Large transformer for multimodal alignment and reasoning	★☆	Extremely High: trillion-token multimodal pretraining	Very High	[9,10,13]
	Cross-modal Pretraining (Contrastive/Masked Modeling)	Unified multimodal alignment via CLIP-style objectives	★★☆	High: paired multimodal data	High	[50–53]
	Token Concatenation + LLM Backbone (PaLM-E, RT-2)	Token-level fusion processed by a multimodal LLM	★☆	Very High: multimodal corpora + robot demos	Very High	[9,10,13]
II. Fusion and Reasoning Module	Cross-Attention Fusion (CoCa, BLIP-2, Flamingo)	Attention-based vision-language fusion	★☆	Very High: paired image-text data	High	[44–47]
	Hierarchical/Planning-Head LLM (VLA-OS, Groot N1)	Multi-head planner for subgoal reasoning	★	Very High: structured embodied datasets	Very High	[63,64]
	Language-Guided Reasoning (SayCan, Inner Monologue)	LLM task planning grounded to low-level controllers	★★☆	Moderate: structured human-labeled task data	Medium	[8]
III. Action Decoder Module	Autoregressive Sequence Generation (RT-series, Gato, OpenVLA)	Transformer decoder generating sequential actions	★★	Moderate: large-scale robot demonstration datasets	Medium-High	[12,74]
	Diffusion Policy (Diffusion-VLA, CogDiff)	Diffusion model for smooth trajectory generation	★	Very High: continuous motion data	High	[79–84]
	Flow Matching (π_0)	ODE-based deterministic trajectory generator	★★★	Moderate: High-simulated + real-world data	High	[86,87,90]
	Mixture-of-Experts Controller (MoLE-VLA)	Parallel experts with dynamic gating	★★	High: diverse robot task data	Very High	[91]
	Self-Correcting Controller (SC-VLA)	Real-time policy refinement with error recovery	★★★	Moderate: task-specific robot data	Medium	[99]
	Energy-Based Control (EBC-VLA)	Optimization-based controller minimizing energy potential	★★	Moderate: state-action pairs or demonstrations	High	[37]

The Action Decoder Module showcases perhaps the most direct trade-offs between inference speed and policy quality. Autoregressive sequence generation methods, exemplified by the RT series and Gato, predict actions token-by-token. This sequential nature inherently limits inference speed; models like RT-1 operate at approximately 3–5 Hz on standard hardware, sufficient for many manipulation tasks but not for highly dynamic control, warranting a "★★" real-time rating [3,12]. Their "Moderate" data requirement is characterized by reliance on large-scale robot demonstration datasets (e.g., the 130,000-trajectory RT-1 dataset [3]), without necessarily requiring additional internet-scale pretraining for the decoder itself. Diffusion-based policies, such as Diffusion Policy, generate actions through an iterative denoising process. While producing smooth and multimodal trajectories, this iterative process (often involving 20–100 steps) results in low inference frequencies, typically 1–2 Hz, leading to a "★" real-time rating [80]. Training these models stably for high-degree-of-freedom control also imposes a "Very High" data requirement, needing extensive datasets of continuous, smooth demonstration trajectories [79,80]. In comparison, flow matching methods (e.g., π_0) present a more efficient alternative. As single-pass, deterministic samplers, they achieve high inference speeds (> 20 Hz), making them suitable for real-time reactive control and earning a "★★★" real-time rating [90]. Their data requirement is assessed as "Moderate-High," benefiting from a mixture of simulated and real-world data without demanding the extreme scale often necessary for training diffusion models [86,90].

This component-wise analysis underscores that the ratings in Table 1 are not arbitrary classifications but are synthesized from reported system performances and resource demands, providing a practical guide for selecting architectural components based on application-specific constraints such as latency tolerance, data availability, and computational budget.

3. The paradigms of VLA architecture

Recent research in VLA systems has increasingly crystallized into two principal architectural paradigms: the end-to-end integrated architecture and the hierarchical archite. These two paradigms embody fundamentally different design philosophies and offer distinct trade-offs in terms of simplicity, interpretability, scalability, and long-horizon task performance. The following sections delineate these two approaches, their respective advantages and limitations, and highlight key recent advancements that are shaping their evolution.

3.1. End-to-end integrated architecture

The end-to-end integrated paradigm employs a single, unified model backbone to process visual inputs (e.g., RGB images, and optionally depth, point-cloud, or ontology states) alongside linguistic tokens within a single sequence. This fused representation is then used to directly decode robot actions, either through autoregressive output generation or via diffusion policies. Pioneering models that exemplify this approach include RT-2 [10], and OpenVLA [12], Open X-embodiment [100], and Octo [101], UniAct [102]. And later works start to explore different model architectures, training objectives, and extra multi-modal representations and information fusion designs to make this paradigm more effective and efficient [103–107]. According to whether the action generation process is conditioned on the planning embeddings or results, they can be further divided into explicit planning and implicit planning. For explicit planning, EmbodiedCoT [108] and CotVLA [109] generate either language-based or goal-image-based embodied chain-of-thought

reasoning before generating actions, and the action generation process is conditioned on the embeddings of CoT. Latent actions refer to compressed, continuous vector representations that encode motor intentions without explicit parameterization of low-level control commands. They serve as an intermediate abstraction between high-level planning and low-level execution, often learned through variational or contrastive objectives. For implicit planning, RoboBrain [110] and ChatVLA [111] train VLA with auxiliary task reasoning loss in language representations. LAPA [112] and LAPO [113] seek to use latent action tokens that serve as forward dynamics representations to generate future images as image foresight planning, and decode these latent actions to real actions with another action head.

Formally, the End-to-End Integrated VLA learns a mapping:

$$(a_t) = \pi_\theta(o_{1:t}, l), \tag{1}$$

where both the observation $o_{1:t}$ (images and proprioception) and the instruction l are tokenized and processed by a multimodal transformer, which is illustrated in Figure 3a. This end-to-end differentiable function implicitly encodes task reasoning and decision-making within the shared latent space. Architecturally, End-to-End Integrated VLA systems typically employ a VLM backbone (such as PaLM-E [9], OpenVLA [12], and Pi0 [90]) with attached action heads that decode either delta-pose vectors [3] or discretized action tokens [78]. Furthermore, some recent implementations utilize flow matching [114,115] or denoising diffusion [80,83,116] objectives to enhance trajectory smoothness and multimodality.

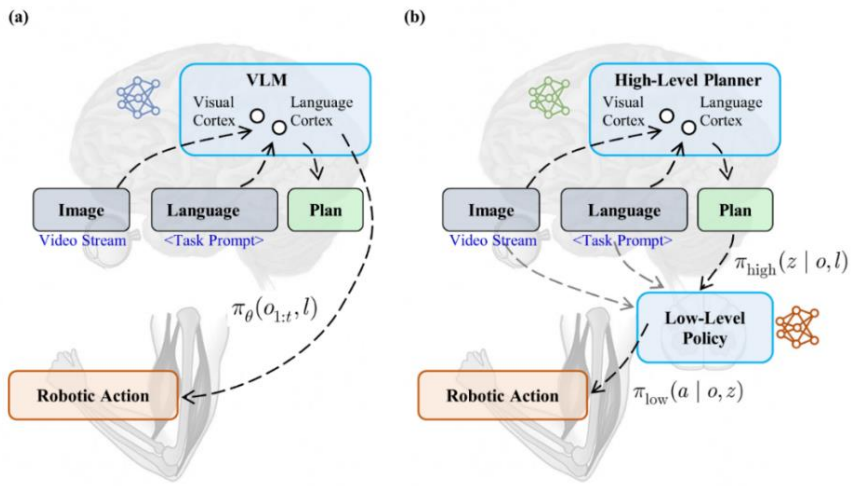

Figure 3. Two different VLA paradigms. **(a)** end-to-end integrated architecture; **(b)** hierarchical architecture.

Researches on End-to-End Integrated-VLA began with early reactive end-to-end models, RT-1 [3] first demonstrated large-scale vision-action learning by training a transformer on over 130,000 human demonstrations, achieving unprecedented diversity in manipulation behaviors. Based on it, RT-2 [10] integrated PaLM-based vision-language pretraining, thereby transferring semantic understanding from web-scale data into embodied control, making a great leap toward grounding abstract language in physical action. PaLM-E [9] unified visual and linguistic embeddings in a single multimodal encoder, enabling it to reason about spatial relations (such as “put the cup on the right of the plate”) unseen during training. This principle of semantic grounding is further developed in CogAct [117], which introduced explicit cognitive attention for aligning vision and language. Similarly, RoboMM [118] consolidate multi-task capabilities within a single large VLA backbone.

Subsequent studies shifted focus toward improving the stability and generalization of end-to-end reasoning. The Gato model [2] by DeepMind laid conceptual foundations by tokenizing heterogeneous modalities (images, proprioception, language) into a unified sequence processed introgressively. Building on this, diffusion policy [80] replaced discrete token prediction with continuous diffusion-based denoising, leading to smoother trajectory generation and improved robustness in multi-step control, then RDT-1B [116] and Dita [119] extended diffusion-based control to bimodal and generalist settings. VLA-0 [120] pushes the paradigm to its logical extreme by demonstrating that a powerful VLA can be constructed without any architectural modifications, simply by prompting a pre-trained VLM to generate continuous robot actions represented as integer text tokens. This "zero-modification" approach underscores the potential of pre-trained multimodal models to serve as effective robotic controllers with minimal adaptation.

Besides, lots of researches are dedicated to embedding implicit planning signals within end-to-end models. CoT-VLA [109] and ECoT [108] introduced the concept of embodied chain-of-thought reasoning, allowing the VLA to generate intermediate goal representations (language-based or goal-image-based) before action prediction. These latent representations serve as internal plan surrogates, thereby implicitly enhancing reasoning depth while maintaining differentiability. Similarly, MDT [121], PIDM [122], and RoboBrain [110] utilized auxiliary losses that encourage the VLA to generate future visual features (image foresight) as a self-supervised planning proxy. All of the techniques mentioned above significantly improved generalization on long-horizon tasks.

Other works have focused on enhancing specific capabilities. DepthVLA [29] extends the integrated architecture for explicit spatial reasoning by fusing semantic and geometric features through shared attention layers in a multi-expert structure. This enables depth-aware reasoning without external 3D supervision, thereby improving manipulation precision and obstacle avoidance within a unified framework. To tackle efficiency concerns, HyperVLA [58] employs a hypernetwork that generates compact, task-specific policies from a large frozen backbone, reducing the number of active parameters by approximately 90 times while maintaining competitive performance, thus enhancing scalability and responsiveness. Additional notable innovations in 2025 further illustrate the paradigm's evolution. FALCON [123] injects spatial tokens directly into the action head rather than the vision-language backbone, preserving semantic alignment while enriching geometric reasoning for unseen physical layouts. X-VLA [87] utilizes soft prompts, which are learnable embeddings that encode embodiment-specific features. This enables a single Transformer backbone to control multiple robot morphologies efficiently. EMMA [124] strengthens visual grounding through diffusion-based visual transfer, generating text-controlled synthetic manipulation videos and using AdaMix re-weighting to boost zero-shot visual generalization. MoLE-VLA [91] dynamically skips layers during inference based on input complexity, reducing latency while maintaining task fidelity for more practical real-time deployment. AdaMoE [92] tackles scalability by integrating a Mixture-of-Experts module into pretrained dense VLAs, achieving efficient specialization and improved real-world performance.

Recent advances such as UniVLA [125], IGOR [126], and video prediction policy (VPP) [127] propose latent action pretraining and video foresight learning, in which the model predicts future frames or latent action sequences as part of its internal reasoning process. These methods demonstrate that end-to-end integrated architectures can approximate hierarchical reasoning within a single unified backbone through learned latent spaces.

This architecture offers several key benefits. Firstly, it enables a unified representation where visual, linguistic, and action concepts are learned jointly, which strongly facilitates zero-shot and few-shot generalization. Secondly, it entails relatively low project overhead as there is no need to engineer explicit interfaces between separate high-level and low-level modules. Finally, its simplicity as a monolithic system makes it easier to implement and maintain compared to more modularized alternatives.

However, the paradigm also presents notable challenges. It often suffers from reduced interpretability, as the model's internal reasoning process is implicit, making it difficult to extract a human-understandable rationale for its decisions. Furthermore, it can exhibit long-horizon instability, where errors accumulate over extended action sequences in the absence of an explicit planning module. Lastly, the resource demands of these large, fused models can lead to slower inference times, particularly with autoregressive decoders, and higher computational and memory costs.

3.2. Hierarchical architecture

In contrast to the monolithic end-to-end approach, the hierarchical architecture explicitly decomposes the problem into separate modules for high-level planning and low-level execution. In this paradigm, a reasoning or planner module first interprets the task and generates a sequence of subgoals or symbolic representations (e.g., textual step-by-step plans, object locations, or SE(3) trajectories). A separate, low-level policy network then executes these subgoals through closed-loop control. This explicit separation enhances interpretability, modularity, and reliability over long-horizon tasks. The idea of hierarchical models has always existed in robotics research [128,129]. Representative systems include SayCan [8], VLA OS [13], Groot N1 [64], and Pi 0.5 [130]. Research like HAMSTER demonstrates that by defining the interface between modules abstractly (e.g., as a 2D path), the high-level planner can leverage diverse off-domain data (e.g., videos and simulation) for cross-embodiment and visual-appearance transfer, achieving substantial performance improvements.

This design reinstates the conceptual separation between symbolic reasoning and motor control while preserving differentiability within each module. The high-level planner often leverages LLMs (such as Qwen2.5 [131], PaLM [9], Gemini [132]) for semantic planning, whereas the policy employs transformer or diffusion-based visuomotor networks [80,90,116]. Formally, this architecture decomposes the control function as Equation (2):

$$\begin{cases} \pi(a \mid o, l) = \pi_{\text{low}}(a \mid o, z) \\ z = \pi_{\text{high}}(z \mid o, l) \end{cases}, \tag{2}$$

where π_{high} denotes the high-level planner generating task representations z according to the instruction l and observation o, and π_{low} denotes the low-level policy executing actions a conditioned on z, which is illustrated in Figure 3b. Therefore, Hierarchical architectures instantiate an embodied cognition hierarchy, aligning closely with cognitive neuroscience models of deliberative and reactive control.

The idea of hierarchical models bridging high-level understanding with low-level execution has always existed in robotics research [133,134]. Action hierarchies are early achieved in SayCan [8] by breaking a long-horizon instruction down to medium horizon tasks.

However, it's one kind of PlanningOnly-VLA [91], the finish of the medium horizon task relies on existing pretrained primitives and often an affordance value function to shape the predications, no short horizon robotic motions can be learnt. Based on the similar thought, a first major step toward hierarchical

decomposition is RT-H [135], which duplicated a VLM backbone into two separate networks: one generating language-based plans, the other executing actions. Subsequent works systematically extended this principle. GR00T N1 [64] introduced a dual-system architecture, which combined a VLM-based reasoning module (System 2) with a diffusion-transformer control module (System 1) to enable co-optimization of high-level semantic planning and low-level motor control across diverse robot embodiments. FAST [78] combined high-level action token generation with a low-level controller trained via flow matching. Hi-Robot [136] demonstrated large-scale open-ended instruction following using a text-based planner that decomposes goals into actionable sub-tasks, whose modular design enabled compositional generalization to unseen verb-object combinations.

Hamster [137] further broadened the paradigm by employing multi-modal plans (2D paths, video foresight, affordance maps) as explicit interfaces, enabling the planner to be pretrained on large non-robotic video datasets, which crucially decoupled high-level reasoning from embodiment constraints, allowing knowledge transfer across embodiments and dynamics. Evo-0 [138] refines this hierarchical design by inserting an implicit spatial reasoning layer that integrates geometry-aware features into the perception-action pipeline, enhancing depth-aware control without explicit 3D sensing. Planning representations are structured intermediate outputs, such as affordance maps, subgoal sequences, trajectory sketches, or symbolic state graphs, that translate high-level task understanding into executable guidance for low-level policies. They enhance interpretability and facilitate hierarchical reasoning. Parallel advancements occurred in visual planning representations. Research such as RT-Affordance [139,140], RoboPoint [141], and Flip [142] designed dense intermediate maps (affordance, keypoints, and sceneflows, respectively) that encode the spatiotemporal semantics of the environment. These representations bridge the symbolic perceptual gap and serve as robust conditioning inputs for policy learning. DexVLA [143] extended this idea by generating visual foresight sequences, where the planner predicts future states as goal images, subsequently refined by the policy network. Recent research such as ManiFoundation [144] applied hierarchical architectures to dexterous hand and bimanual control, proving the adaptability to complex embodiments.

Recent research has significantly enriched the hierarchical paradigm. A prominent example is OneTwoVLA [145], which combines an autoregressive “plan generator” with a diffusion-based “executor.” The high-level Transformer interprets multimodal instructions to predict symbolic subgoals, while the low-level diffusion policy refines these into executable trajectories, effectively bridging symbolic planning with continuous motion control. NoTVLA [146] extends the hierarchy with a non-tokenized architecture, encoding both semantic goals and low-level dynamics as continuous latent vectors. This enables smoother information flow between planning and control, improving both accuracy and inference speed compared to token-based systems. Gemini Robotics 1.5 [147] unites a multi-embodiment control model with an embodied reasoning module, facilitating motion transfer across different robots and interleaved “thinking-acting” cycles for robust long-horizon performance. VLA ^{2}[60] couples two interconnected VLA modules for strategic reasoning and execution, incorporating emergent self-verification where the reasoning layer evaluates and corrects the control layer’s outputs before execution. PhysiAgent [67] integrates embodied physics simulation into the reasoning loop, allowing an LLM planner to evaluate candidate plans for physical feasibility through a differentiable physics predictor. Other works seek to generate multi-modal planning results for policy learning, such as image flows or trajectories [148], future videos [149,150], and keypose [151].

The hierarchical paradigm offers several compelling advantages. It provides enhanced interpretability, as the high-level planner produces human-readable decisions, making it easier to trace and diagnose errors. It also delivers better long-horizon stability by explicitly breaking down complex tasks, thereby reducing error accumulation across long action sequences. Furthermore, it facilitates cross-domain generalization because abstract planning interfaces allow the system to leverage more diverse data sources and transfer knowledge across different robot embodiments effectively. Finally, it enables error isolation, where failures in planning are contained at the interface level without corrupting the representations of the low-level control policy.

The primary drawbacks of this approach involve increased engineering overhead, as the individual modules and their interfaces must be carefully designed and integrated. This can also lead to potential latency due to the computational overhead of multiple modules and bottlenecks in inter-module communication. Moreover, the modularity of hierarchical systems, while beneficial for interpretability and long-horizon stability, inherently sacrifices some of the "end-to-end" simplicity and the benefits of a fully unified representation, such as seamless gradient propagation and joint representation learning, found in the integrated paradigm.

3.3. Design principles for VLA architecture selection

3.3.1. Criteria for architectural selection

End-to-end integrated architectures are preferable under conditions characterized by short-to-medium task horizons (typically 5–20 action steps) where error accumulation can be managed through closed-loop feedback (as shown in Table 2). These architectures excel at reactive control and continuous manipulation but struggle with tasks requiring deep sequential reasoning or multi-stage decomposition. They are particularly suited for scenarios demanding strong zero-shot or few-shot generalization to novel objects, environments, or linguistic instructions, as their unified representation space facilitates cross-modal transfer from pre-trained VLMs. This paradigm benefits significantly from scale, requiring the availability of large-scale, diverse demonstration datasets for joint training of perception and control modules. It is favored when development simplicity, minimal integration overhead, and end-to-end differentiability are prioritized over interpretability or long-horizon robustness. Typical applications include open-vocabulary object manipulation, visual servoing, reactive navigation, and tasks where semantic grounding is more critical than explicit planning.

Conversely, hierarchical architectures are better suited for long-horizon, compositional tasks, such as "prepare a meal" or "assemble furniture," that require explicit decomposition into subgoals and sequential reasoning. The explicit separation of planning and execution in this paradigm effectively mitigates error propagation. It is essential for applications where human oversight, failure diagnosis, regulatory compliance, or ethical considerations demand transparent decision-making, as seen in healthcare robotics, human-robot collaboration, and industrial safety-critical systems. The hierarchical design facilitates cross-embodiment and transfer learning by enabling a single high-level planner to interface with diverse low-level controllers across different robot morphologies, sensor configurations, or hardware platforms. This architecture is also advantageous in scenarios of data heterogeneity and scarcity, as it can leverage non-robotic data sources, including web videos, textual instructions, and simulation trajectories, to train the high-level planner, thereby overcoming the scarcity of robot demonstration data.

Typical applications encompass household chore automation, autonomous assembly lines, scientific experimentation, and any scenario where task complexity exceeds the reliable horizon of reactive policies.

Table 2. The two paradigms of VLA architecture.

Paradigms	End-to-End Integrated Architecture	Hierarchical Architecture
Core Idea	Employs a single unified multimodal backbone that jointly encodes vision, language, and action to directly predict control outputs.	Explicitly decomposes reasoning and control into a high-level planner and a low-level executor, connected through intermediate task representations.
Representative Models	RT-1 [3], RT-2 [10], PaLM-E [9], Gato [2], Diffusion Policy [80]	SayCan [8], VLA-OS [13], RT-H [135], Gemini Robotics 1.5 [147], VLA ²[60].
Model Structure	Single multimodal Transformer or diffusion model with task-specific action heads.	Dual (or multi-module) structure: a high-level reasoning module (often an LLM-based planner) and a low-level visuomotor or diffusion-based policy network.
Advantages	Unified multimodal representation; Low integration overhead; Strong zero/few-shot generalization; Fully differentiable and easy to scale	High interpretability via explicit plans; Stable long-horizon reasoning; Modular and transferable across embodiments; Errors isolated between modules
Limitations	Implicit, non-transparent reasoning; Error accumulation in long-horizon tasks; High compute and latency for large models; Weaker control over internal abstractions	Higher engineering and interface design cost; Inter-module communication latency; Less synergy from end-to-end optimization; Requires careful interface tuning
Suited Scenarios	Open-vocabulary and short- to mid-horizon manipulation requiring flexible generalization and large-scale multimodal learning.	Long-horizon, compositional, or multi-stage tasks that demand interpretable reasoning, modular control, and cross-domain transfer.

An emerging trend is the development of hybrid or adaptive architectures that dynamically combine end-to-end and hierarchical elements to preserve representational flexibility while incorporating interpretability and long-horizon robustness. These systems employ conditional switching mechanisms triggered by task progress or uncertainty, utilize multi-level representation learning to maintain both visuomotor and symbolic abstractions, and explore learnable interfaces between modules through self-supervised objectives. This convergence aims to create more robust and generalizable embodied agents capable of operating across a wide spectrum of task complexities and environmental conditions, representing a pivotal direction for next-generation VLA systems.

3.3.2. Optimization of LLM integration depth in planning

The integration depth of LLMs within the planning loop constitutes a critical design dimension that mediates between reasoning capacity and operational constraints such as latency and computational cost. We categorize LLM involvement into three progressively deeper tiers, each characterized by distinct functional roles, performance trade-offs, and application scenarios.

The first tier, lightweight LLM conditioning, primarily delegates to the LLM the role of instruction parsing and task categorization. Here, natural language commands are mapped to predefined skill labels or symbolic goal representations, such as “pick,” “place,” or “navigate to,” which subsequently condition a downstream policy. This approach minimizes inference latency and computational overhead, facilitating straightforward integration with existing control pipelines. Its limitations, however, include restricted compositional reasoning, an inability to generalize to novel verb-object combinations, and a dependence on preconstructed skill libraries. Representative instantiations include RT-1 and the task-conditioning mode of Gato. This tier is well-suited for structured environments with

constrained task vocabularies, real-time control applications, and systems operating under stringent computational budgets.

The second tier, moderate LLM-based planning, elevates the LLM to a subgoal-generation role. The model decomposes instructions into stepwise subgoals expressed in natural language or structured formats, for example, “first locate the red cup, then grasp it, finally move to the table,” thereby engaging in basic reasoning about object properties, spatial relations, and action sequences. This capability supports compositional generalization and enables recovery from errors through re-planning, albeit at the cost of increased inference time (typically in the range of 100–500 ms per planning step). Challenges include the potential generation of physically infeasible subgoals and a reliance on robust grounding mechanisms to translate symbolic plans into executable actions. Systems such as SayCan, VLA-OS, and RT-H exemplify this tier, which finds application in medium-complexity manipulation tasks, interactive robotics, and scenarios where a degree of explainability is required.

The third and deepest tier, deep LLM reasoning, entrusts the LLM with symbolic and physical reasoning, often augmented by chain-of-thought prompting, internal search algorithms (e.g., Monte Carlo Tree Search), or interaction with simulation engines. In this mode, the model evaluates plan feasibility, predicts physical outcomes, and performs metacognitive monitoring to self-correct based on environmental feedback. Such deep integration affords robust operation in unstructured environments, sophisticated causal reasoning, and advanced failure recovery. These benefits, however, come with substantial computational overhead, often requiring seconds to minutes per decision, as well as increased system complexity and difficulty in maintaining real-time responsiveness. Representative implementations include PhysiAgent, VLA-Reasoner, and frameworks employing embodied chain-of-thought reasoning. This tier is justified in autonomous systems operating in highly dynamic or novel environments, scientific discovery robots, and tasks where pre-trained skills are inadequate for novel situations.

A guiding principle for selecting the appropriate integration tier is that the required depth of LLM involvement increases with (1) task complexity and horizon, (2) environmental uncertainty and dynamism, (3) demands for explainability and safety, and (4) the availability of computational resources for inference-time reasoning. Consequently, real-time control applications, such as drone navigation or surgical robotics, typically necessitate Tier-1 or Tier-2 integration, whereas slower-paced, cognitively demanding tasks like experimental design or long-term autonomy may warrant the deeper reasoning capabilities of Tier-3.

3.3.3. Toward hybrid and adaptive architectures

The complementary strengths and limitations of end-to-end and hierarchical paradigms suggest that hybrid and adaptive designs are increasingly explored and appear promising as a means to leverage the fundamental trade-offs between reactive speed and deliberative reasoning. This trajectory represents not merely a stacking of modules, but a deep fusion realized through several key design strategies.

First is the introduction of Asynchronous Hierarchical Control Topologies. To resolve the fundamental conflict between the inference latency of large models and the real-time response requirements of robotics, hybrid architectures typically adopt a dual-loop structure: an upper-level “Thinking” loop (typically driven by an LLM) operates at a lower frequency (e.g., 0.5 Hz) to perform long-horizon reasoning and update high-level goals, while a lower-level “Acting” loop (typically driven by a diffusion or flow-matching policy) runs at a high frequency (e.g., 20 Hz or higher) to track and

execute these goals via an asynchronous execution mechanism. This decoupled design preserves high-level semantic planning capabilities while ensuring low- level agility and closed-loop stability.

Second is the development of conditional switching mechanisms, which enable systems to dynamically adjust their computational pathways based on task uncertainty or complexity. For instance, when handling routine or well-practiced motions, the system maintains efficient end-to-end reactive control to minimize latency. However, upon encountering novel objects, complex logic, or detecting execution deviations, it dynamically activates hierarchical planning modules to intervene and replan.

Third is multi-level representation learning. To balance generalization with precision, models are trained to simultaneously preserve low-level visuomotor abstractions (for reactive control) and high-level symbolic abstractions (for deliberative planning). Concurrently, researchers are exploring Learnable Interfaces, training the communication channels between modules via self-supervised objectives to minimize the semantic gap and engineering overhead typical of traditional modular systems.

Furthermore, unified backbones with specialized heads offer a pathway toward architectural convergence. A single differentiable model can support both direct action generation and explicit subgoal or plan prediction, thereby blending the advantages of both paradigms within a cohesive framework.

Recent works have empirically validated this convergence trend. For example, OneTwoVLA [145] combines an autoregressive plan generator with a diffusion-based executor, effectively bridging symbolic planning with continuous motion control. Gemini Robotics 1.5 [147] integrates multi-embodiment control with interleaved “thinking-acting” reasoning cycles, demonstrating robust long-horizon performance. By strategically blending these architectural elements, hybrid designs aim to achieve the representational flexibility and sample efficiency of end-to-end learning alongside the structured reasoning, interpretability, and long-horizon robustness of hierarchical systems. This comprehensive paradigm is poised to overcome the core limitations of singular approaches, paving the way for more generalizable and deployable embodied agents.

3.4. Temporal vs. precision trade-offs: a quantitative perspective

In robotic systems, the fundamental design tension between real-time responsiveness (latency) and task execution accuracy (spatial precision, success rate) is concretely manifested in the performance profiles of end-to-end and hierarchical VLA architectures. These paradigms occupy distinct regions in the latency-accuracy design space, governed by their intrinsic computational flow, reasoning mechanisms, and modular organization. This section provides a structured, evidence-based comparison of the two approaches along these two critical axes, synthesizing quantitative metrics reported in the literature to elucidate their inherent trade-offs.

3.4.1. Real-time performance and latency

Real-time capability, typically quantified as control frequency (Hz) or perception-to-action latency (ms), is predominantly determined by model complexity, inference parallelism, and the sequential depth of the reasoning pipeline.

End-to-end integrated architectures are optimized for low-latency, single-step inference. By processing multimodal inputs and generating control outputs through a single, differentiable forward pass, they minimize inter-module communication overhead. Lightweight instantiations, such as the

original RT-1 model, achieve stable control frequencies of 3–5 Hz on physical hardware, enabling responsive closed-loop manipulation. Subsequent optimizations through efficient decoders, such as the flow-matching approach in π_0, can elevate this to 10–20 Hz or beyond for tasks where high-frequency reaction is critical. However, this advantage is highly contingent on model scale. Architectures that integrate massive pre-trained vision-language backbones, such as PaLM-E and RT-2, incur significantly higher per-step inference times, often exceeding 100−500 ms, due to the computational burden of their billion-parameter transformers, thereby constraining their applicability in highly dynamic, safety-critical scenarios.

Hierarchical architectures, in contrast, inherently introduce higher latency due to their staged, deliberative processing. The operational cycle requires sequential execution of: (1) high-level task decomposition and planning (often by a large LLM), (2) translation of symbolic plans into controller-compatible representations, and (3) execution by a low-level visuomotor policy. Each stage contributes additive delay. For instance, the LLM-based planner in systems like SayCan or VLA-OS commonly requires 100–500 ms for inference, and when augmented with search-based reasoning (e.g., Monte Carlo Tree Search in VLA-Reasoner or physics simulation (e.g., PhysiAgent), planning latency can extend to several seconds. Although the low-level policy itself may be computationally efficient (e.g., a high-frequency diffusion or flow policy), the cumulative planning-execution loop typically operates at 1–2 Hz for complex instructions. This makes hierarchical systems less suited for tight, reactive control loops but effective for tasks where deliberation outweighs speed. A common mitigation strategy is asynchronous execution, where the high-level planner operates on a slower cycle, updating a shared goal buffer that a faster, reactive low-level policy consumes.

The dominant factors governing latency are model size and architectural complexity, the depth of inference-time reasoning, the choice of action decoding paradigm (autoregressive, diffusion, flow), and, for hierarchical systems, the overhead associated with inter-module communication and representation translation.

3.4.2. Task accuracy and spatial precision

Accuracy encompasses both overall task success rate on standardized benchmarks and fine-grained spatial precision (e.g., millimeter-level positioning error, success in dexterous manipulation). This dimension is closely linked to a system's capacity for explicit geometric reasoning, state awareness, and robustness to error propagation.

End-to-end integrated architectures demonstrate exceptional zero-shot and few-shot generalization capabilities, often achieving high success rates when confronted with novel objects, background scenes, or paraphrased instructions. This strength is derived from their unified representation learning on internet-scale data. For example, RT-2 shows remarkable performance on open-vocabulary manipulation tasks unseen in its robotic training data. However, this strength is counterbalanced by notable weaknesses in long-horizon compositional accuracy. The absence of explicit state representation and planning leads to error accumulation over extended action sequences, causing gradual drift or catastrophic failure in multi-stage tasks. Furthermore, their spatial precision is often limited, as metric reasoning must be implicitly learned from pixel correlations rather than explicitly computed. Consequently, tasks demanding sub-centimeter precision or precise geometric relationships, such as

“insert the USB drive,” reveal limitations in their underlying spatial grounding, as highlighted in critical analyses.

Hierarchical architectures are architecturally predisposed towards high accuracy in long-horizon, structured tasks. By explicitly decomposing problems into subgoals, they naturally contain error propagation: a failure in one sub-step can often be detected and recovered without derailing the entire mission. This leads to superior performance on benchmarks designed for compositional reasoning, such as RoboHiMan. When the interface between planner and executor is geometrically grounded, for example, using affordance maps, keypoint trajectories, or SE(3) goal poses, hierarchical systems can achieve high spatial precision. The low-level policy can be trained specifically for metric control using targeted demonstration data, avoiding the abstraction gap present in pixel-to-action mapping. Moreover, the modular design facilitates the integration of dedicated verification modules, safety filters, and reflexive recovery policies, such as FailSafe and SEAL. These components can monitor execution in real-time, detect anomalies, and trigger corrective actions, thereby systematically enhancing overall task reliability and robustness in uncertain environments.The primary determinants of accuracy are the explicitness of reasoning and state representation, the degree of geometric and physical grounding in the perceptual and planning representations, the specificity and quality of training data for the control policy, and the presence of active error detection and recovery mechanisms.

3.4.3. Synthesis and guiding principles

The quantitative interplay between latency and precision reveals a clear, complementary specialization between the two architectural paradigms. End-to-end integrated models excel in scenarios demanding low-latency, adaptive response and broad semantic generalization over shorter time horizons, albeit with potential compromises in long-horizon reliability and metric precision. Hierarchical models are the preferred choice for applications requiring deliberative reasoning, high long-horizon success rates, and verifiable spatial accuracy, accepting higher per-decision latency as the cost for this structured robustness.

This analysis yields a practical guiding principle (as shown in Table 3): the selection of a VLA architecture is not merely a choice of model but a strategic allocation of computational resources along the latency-precision frontier. For applications such as human-robot collaboration in dynamic settings or agile navigation, where milliseconds matter and tasks are often short-term, end-to-end architectures provide a compelling solution. For applications such as autonomous assembly, precision logistics, or complex task learning from demonstration, where correctness, safety, and multi-step reasoning are paramount, hierarchical architectures offer the necessary framework. The emerging frontier of hybrid and adaptive systems seeks to dynamically navigate this trade-off, employing end-to-end reactivity for routine operations while invoking hierarchical deliberation for novel or critical subtasks, thereby aiming to capture the high-performance regions of both paradigms.

Table 3. Architectural selection.

Task/System Condition	Recommended Base Architecture	Time Performance (Typical)	Spatial Error (Typical)	Role of Planner/LLM	Design Rationale	Representative Works
Short-horizon tasks (< 20 steps), real-time control	End-to-End	Control Frequency: 3–20 Hz Latency: 50–300 ms	~1–3 cm	None or Tier-1 (instruction conditioning)	Reactive visuomotor policies provide sufficient stability and minimal latency	RT-1; Gato; OpenVLA
Medium-horizon manipulation with novel instructions	End-to-End + Latent Planning	Control Frequency: 1–5 Hz Planning Latency: 100–500 ms (per subgoal)	~2–5 cm	Tier-2 (latent subgoal or skill guidance)	Latent planning improves compositional generalization without heavy planning overhead	CoT-VLA; ECoT; RoboBrain; MDT
Long-horizon, multi-stage tasks	Hybrid (Reactive policy + Explicit Planner)	Planning Frequency: 0.2–2 Hz Execution Frequency: 5–20 Hz (low-level)	< 1 cm (if low-level policy is fine-tuned)	Tier-2 (conditional subgoal planning)	Sparse planning mitigates error accumulation while preserving execution speed	SayCan; RT-H; VLA-OS
High uncertainty or failure-prone environments	Adaptive Hybrid	Nominal Frequency: 3–10 Hz Recovery Latency: 1–5 s (on invocation)	Variable	Tier-2/3 activated on demand	Planner invoked only upon uncertainty or failure to reduce constant latency	PhysiAgent; INSIGHT; FailSafe
Safety-critical or human-facing applications	Hierarchical or Hybrid	Planning Frequency: 0.5–1 Hz Verification Overhead: +100–500 ms	Can be very low (< 0.5 cm) with geometric interfaces	Tier-2 with interpretable plans	Explicit reasoning and subgoals support transparency and verification	SayCan; VLA-OS; SEAL
Limited robot data, heterogeneous sources	Hybrid	Planning Frequency: 0.5–2 Hz Data Efficiency: High (leverages non-robot data)	Dependent on low-level policy's training data	Tier-2 leveraging web/simulation data	Planner generalizes from non-robot data, reducing demonstration requirements	PaLM-E; RT-2; VLA [2]

4. Discussion and outlook

The rapid evolution of VLA architectures has demonstrated the feasibility of unified perception-reasoning-action systems, yet significant challenges remain before robust real-world deployment. This section synthesizes the core limitations of current approaches and outlines promising research directions.

4.1. A structured analysis of failure modes in VLA architectures

Despite their promising capabilities, current VLA systems exhibit recurring and systematic failures that reveal fundamental gaps in their design. A structured analysis, organized by failure type and underlying architectural cause, is essential for diagnosing limitations and guiding future research. We categorize prevalent failures into three interconnected domains: semantic and spatial grounding, long-horizon compositional reasoning, and sim-to-real generalization.

Semantic and spatial grounding failures often manifest as misinterpretations of spatial relations or object attributes. For example, an instruction like "place the red cup to the right of the blue bowl" may result in a placement merely near the bowl, losing the precise relational intent. This failure is most acute in end-to-end models trained primarily on robot data (e.g., early versions of RT-1 and VIMA), where the alignment between language and visual geometry is learned implicitly and remains shallow. The root cause is the dissociation between the broad semantic knowledge acquired from web-scale pretraining and the metric, egocentric spatial reasoning required for accurate physical interaction. Even models with

explicit 3D perception, such as DepthVLA and EmbodiedGPT, can fail due to errors in projecting 2D semantic features into a consistent 3D scene understanding, leading to misreaches or collisions.

Long-Horizon compositional failures occur when models must execute or plan extended sequences of interdependent actions, such as in "make a sandwich" or "clear the table and wash the dishes." End-to-end integrated architectures (e.g., Gato, RT-2) are particularly susceptible due to error accumulation in their autoregressive action sequences and the lack of an explicit internal representation of subgoals. Their failure mode is often gradual drift or irrecoverable execution errors midway through a task. In contrast, hierarchical architectures (e.g., SayCan, VLA-OS) fail differently: their high-level planner may produce a logically sound subgoal sequence, but a poorly defined or lossy interface between the planner and the low-level policy can render subgoals infeasible or misinterpreted. This planner-executor mismatch highlights that modularity alone does not guarantee robustness; the representational alignment across modules is critical.

Sim-to-Real and generalization failures expose the fragility of models when faced with distribution shifts. Policies trained predominantly in simulation, for example, many diffusion-based policies like Diffusion-VLA, frequently degrade under real-world variations in lighting, texture, and object dynamics. This points to a perception-action gap where visual features used for control are not sufficiently invariant to domain changes. Furthermore, catastrophic forgetting during fine-tuning, where adapting a pre-trained VLA to a new skill severely degrades its performance on original tasks, is a significant issue for methods like VLM2VLA. This indicates that current parameter-efficient tuning strategies may not adequately protect consolidated knowledge, limiting lifelong learning.

Cross-Modal and Interactive Reasoning Failures represent a higher-order challenge. Most current VLAs operate in a single forward pass, unable to ask clarifying questions or recognize their own uncertainty when instructions are ambiguous. This passive processing can lead to the execution of incorrect but plausible interpretations. Systems also struggle with dynamic physical reasoning, such as predicting the outcome of pushing a stack of objects or handling deformable materials, as seen in the limitations of purely vision-based models in tasks requiring force feedback. Integrating tactile and torque sensing, as in MLA and torque-aware vision-language-action (TA-VLA), is a step forward, but fusing these heterogeneous, high-bandwidth sensory streams into a coherent state representation remains an open problem.

In summary, failures are not random but systematically linked to architectural choices (as shown in Figure 4). End-to-end models struggle with long-horizon planning, while hierarchical models introduce interface latency. These insights demonstrate that hybrid designs are increasingly promising, as they leverage structured reasoning to maintain consistency in complex, multi-stage missions without sacrificing reactive agility.

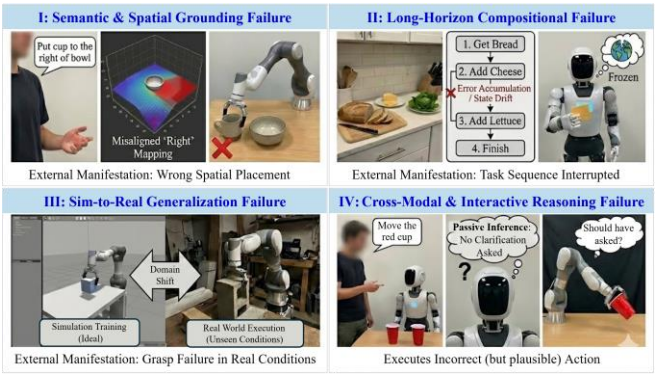

Figure 4. Four major failure cases.

4.2. Critical gaps in current evaluation benchmarks

Current evaluation benchmarks for VLA systems, such as Robot Hierarchical Manipulation evaluation paradigm (RoboHiMan) [152] and testing and evaluating vision-language-action (VLATest) [153], provide valuable frameworks for quantifying performance across standardized tasks. However, these benchmarks exhibit significant methodological limitations that obscure their ability to predict real-world robustness and generalizability. A critical analysis reveals five principal dimensions inadequately captured by existing evaluation paradigms.

Open-World adaptability. Existing benchmarks predominantly assess models in controlled, static environments characterized by predefined object sets, consistent lighting conditions, and minimal scene variation. Consequently, they fail to evaluate a model's capacity to adapt to open-world conditions, such as handling unseen object configurations, novel material properties, or unexpected human interventions. These capabilities are indispensable for deployment in unstructured settings like homes, hospitals, or warehouses.

Long-Horizon robustness under distribution shift. While benchmarks measure task success rates within training-like environments, they seldom probe performance degradation under systematic distribution shifts. Critical real-world variations, including changes in camera viewpoint, illumination, sensor noise, or object dynamics, are rarely incorporated into evaluation protocols. This creates a substantial gap in assessing sim-to-real transfer viability.

Human-in-the-Loop interactivity. Prevailing evaluation methodologies operate largely in an open-loop manner, assuming single, unambiguous instructions are provided ex ante. They neglect essential interactive capabilities, such as a robot's ability to request clarification, incorporate mid-task corrections, or recover from misunderstandings through iterative dialogue. These functions are fundamental to collaborative and instructional real-world scenarios. Spatial reasoning consistency *versus* pattern matching. Success metrics in current benchmarks often prioritize task completion over the quality of underlying reasoning. This obfuscates the distinction between genuine geometric or relational understanding and superficial pattern matching. For instance, a model might successfully "place the cup to the right of the bowl" by recalling correlated spatial arrangements from training data rather than inferring the intended spatial relation de novo which is a failure of reasoning that benchmarks typically do not expose or penalize.

Systemic data and generalization biases. Underlying many of these measurement gaps are fundamental data-related challenges. The scarcity of large-scale, high-quality, multi-modal robot demonstration data, combined with significant heterogeneity across platforms, tasks, and sensor configurations, creates inherent biases in training and evaluation. Benchmarks constructed from such limited data fail to measure a model's true generalization potential. To mitigate this, future benchmarks must be designed with greater task diversity, multi-modal data alignment, and systematic stress-testing of cross-embodiment generalization, the ability for a policy trained on one robot platform to adapt to another. Furthermore, evaluation protocols should consider the data sources used for training, for example, simulation *versus* real-world, web-scale *versus* domain-specific, and their impact on measured performance, as emphasized by studies like VLATest [153] which highlight sensitivity to environmental variations.

These identified gaps, compounded by underlying data limitations, underscore the pressing need for next-generation benchmarks. Such frameworks must incorporate multi-environment testing suites,

adversarial scenario generation, interactive human-in-the-loop evaluation protocols, and explicit metrics for assessing spatial, causal, and compositional reasoning. Equally important is the establishment of systematic and uniform training and evaluation metrics, along with open, unified, and large-scale evaluation benchmarks that account for model robustness across diverse data sources and embodiment platforms. Only through such comprehensive, stress-testing evaluation frameworks can the field advance toward VLA systems that are truly robust, interpretable, and deployable in open-world settings.

4.3. Core challenges and future directions

Current VLA models primarily rely on textual instructions, yet real-world human-robot interaction depends on richer multi-modal inputs [154]. Future research must first broaden the scope of visual signals, extending from 2D images to point clouds, depth maps, high-frame-rate video streams, and beyond, to provide more comprehensive environmental perception. Second, it is essential to integrate non-visual modalities such as tactile and force/torque feedback. For instance, MLA [56] fuses visual, point-cloud, and tactile data, while TA-VLA [155] incorporates joint torque feedback to enhance understanding of physical interaction and enable "force-sensitive" control. A core challenge lies in designing more effective tokenization and alignment mechanisms for these heterogeneous modal inputs to achieve seamless cross-modal fusion. Furthermore, within the language modality, deep integration of audio perception should be pursued to enable fast, continuous speech understanding, allowing robots to engage in full dialogue interactions and be interrupted in real time by new voice commands. Studies such as VLAs [156] represent important explorations in this direction.

As multi-modal perception and action generation capabilities continue to advance, the application scenarios for VLAs are expected to expand significantly. Beyond current real-time manipulation tasks, VLA technology is anticipated to play a central role in industrial precision assembly, personalized domestic services, medical rehabilitation assistance, and operations in extreme environments such as space and deep sea. For instance, the hierarchical benchmark framework proposed by RoboHiMan systematically promotes compositional generalization and robust execution in long-term complex tasks such as navigation, assembly, and organization, highlighting the potential direction of VLAs in driving the development of general embodied agents [157]. Across these diverse applications, safety and reliability remain paramount, requiring rigorous safety verification mechanisms and intrinsically safe constraint designs to ensure that all actions avoid unintended harm to the environment and humans.

5. Conclusion

VLA architectures have demonstrated that unifying language, vision, and action within differentiable frameworks can overcome the brittleness of modular pipelines, enabling robots to generalize across task variations and handle natural language instructions. However, critical challenges remain unresolved. End-to-end models, while achieving impressive zero-shot generalization on internet-scale data, struggle with spatial precision and long-horizon consistency. These failures emerge from implicit rather than explicit reasoning. Hierarchical models provide interpretability and stability through explicit planning stages but sacrifice the representational flexibility that enables broad generalization. Therefore, synthesizing the complementary strengths and limitations of these paradigms, hybrid and adaptive designs are increasingly explored and appear promising. By leveraging the trade-offs between reactive

control and long-horizon reasoning, these architectures offer a balanced path to overcome the bottlenecks of latency and compositional failures. Beyond architectural design, the field must address three pressing challenges: (1) developing evaluation protocols that test genuine spatial reasoning rather than pattern matching, (2) bridging the sim-to-real gap for manipulation tasks requiring sub-centimeter precision, and (3) establishing theoretical frameworks to predict when end-to-end learning will succeed *versus* when explicit structure is necessary. Until these questions are resolved, VLA deployment will remain limited to tasks where approximate solutions suffice.

Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors used generative AI tools only to improve language and readability. The authors take full responsibility for the content of the manuscript.

Acknowledgments

This work was supported in part by the Research Grants Council of Hong Kong (Ref. No. 14204423), in part by the Innovation and Technology Fund (ITF), Guangdong–Hong Kong Technology Cooperation Funding Scheme (Project No. GHP/234/23GD), in part by Shenzhen Science and Technology Project (ZDCY20250901101959006), in part by the Guangdong Provincial Science and Technology Program under Grant 2025A0505080012.

Authors' contribution

Conceptualization, visualization, writing—original draft preparation, Jingjing Pei, Xiaoyin Zheng, Yang Liu, Bike Zhu and Daifeng Wang; writing—review and supervision, Jiajun An, Richard Voyles and Xin Ma; funding acquisition, Xin Ma. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

The authors declare no conflict of interest.

References

[1] Zhao T, Kumar V, Levine S, Finn C. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv* 2023, arXiv:2304.13705.

[2] Reed S, Zolna K, Parisotto E, Colmenarejo SG, Novikov A, *et al.* A generalist agent. *arXiv* 2022, arXiv:2205.06175.

[3] Brohan A, Brown N, Carbajal J, Chebotar Y, Dabis J, *et al.* RT-1: robotics transformer for real-world control at scale. *arXiv* 2022, arXiv:2212.06817.

[4] Naveed H, Khan AU, Qiu S, Saqib M, Anwar S, *et al.* A comprehensive overview of large language models. *ACM Trans. Intell. Syst. Technol*. 2025, 16(5):1–72.

[5] Chang Y, Wang X, Wang J, Wu Y, Yang L, *et al.* A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol*. 2024, 15(3):1–45.

[6] Ma Y, Song Z, Zhuang Y, Hao J, King I. A survey on vision-language-action models for embodied ai. *arXiv* 2024, arXiv:2405.14093.

[7] Kawaharazuka K, Oh J, Yamada J, Posner I, Zhu Y. Vision-language-action models for robotics: a review towards real-world applications. *IEEE Access* 2025,13:162467–162540.

[8] Ahn M, Brohan A, Brown N, Chebotar Y,Cortes O, *et al.* Do as I can, not as I say: grounding language in robotic affordances. *arXiv* 2022, arXiv:2204.01691.

[9] Driess D, Xia F, Sajjadi MS, Lynch C, Chowdhery A, *et al.* PaLM-E: an embodied multimodal language model. *arXiv* 2023, arXiv:2303.03378.

[10] Zitkovich B, Yu T, Xu S, Xu P, Xiao T, *et al.* Rt-2: vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the Conference on Robot Learning*, Atlanta, USA, November 6–9, 2023, pp. 2165–2183.

[11] JiangY, Gupta A, Zhang Z, Wang G, Dou Y, *et al.* VIMA: robot manipulation with multimodal prompts. *arXiv* 2023, arXiv:2210.03094.

[12] Kim MJ, Pertsch K, Karamcheti S, Xiao T, Balakrishna A, *et al.* OpenVLA: an open-source vision-language-action model. *arXiv* 2024, arXiv:2406.09246.

[13] Gao C, Liu Z, Chi Z, Huang J, Fei X, *et al.*VLA-OS: structuring and dissecting planning representations and paradigms in vision-language-action models. *arXiv* 2025, arXiv:2506.17561.

[14] Lu G, Guo W, Zhang C, Zhou Y, Jiang H, *et al.* VLA-RL: towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv* 2025, arXiv:2505.18719.

[15] Han K, Wang Y, Chen H, Chen X, Guo J, *et al.* A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* 2022, 45(1):87–110.

[16] Chen Z, Xie L, Niu J, Liu X, Wei L, *et al.* Visformer: the vision-friendly transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, Montreal, Canada, October 11–17, 2021, pp. 589–598.

[17] Li B, Hu Y, Nie X, Han C, Jiang X, *et al.* Dropkey for vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada, June 18–22, 2023, pp. 22700–22709.

[18] MaoX, Qi G, Chen Y, Li X, Duan R, *et al.* Towards robust vision transformer. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, New Orleans, USA, June 18–24, 2022, pp. 12042–12051.

[19] Woo S, Debnath S, Hu R, Chen X, Liu Z, *et al.* Convnext V2: co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Canada, June 18–22, 2023, pp. 16133–16142.

[20] Yu W, Zhou P, Yan S, Wang X. Inceptionnext: when inception meets convnext. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, Seattle, USA, June 17–21, 2024, pp. 5672–5683.

[21] Li X, Ding H, Yuan H, Zhang W, Pang J, *et al.* Transformer-based visual segmentation: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 2024, 46(12):10138–10163.

[22] Man Y, Gui L, Wang Y. Situational awareness matters in 3D vision language reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, USA, June 17–21, 2024, pp. 13678–13688.

[23] Zhai H, Zhang X, Zhao B, Li H, He Y, *et al.* Splatloc: 3D gaussian splatting-based visual localization for augmented reality. *IEEE Trans. Vis. Comput. Graphics* 2025, 31(5):3591–3601.

[24] Zhou J, Wang J, Ma B, Liu Y, Huang T, *et al.* Uni3D: exploring unified 3d representation at scale. *arXiv* 2023, arXiv:2310.06773.

[25] Mu Y, Zhang Q, Hu M, Wang W, Ding M, *et al.* Embodiedgpt: vision-language pre-training via embodied chain of thought. In *Proceedings of the Conference on Neural Information Processing Systems*, New Orleans, USA, December 10–16, 2023, pp. 25081–25094.

[26] Qi CR, Yi L, Su H, Guibas LJ. Pointnet++: deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the Conference on Neural Information Processing Systems*, Long Beach, USA, December 4–9, 2017, pp. 5105–5114.

[27] Li Z, Wang W, Li H, Xie E, Sima C, *et al.* Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Trans. Pattern Anal. Mach. Intell*. 2024, 46(1):1–14.

[28] Sun B, Cadena C, Pollefeys M, Blum H. OpenFrontier: general navigation with visual-language grounded frontiers. In *IROS 2025 Workshop: Open World Navigation in Human-Centric Environments*, Hangzhou, China, October 19–25, 2025.

[29] Yuan T, Liu Y, Lu C, Chen Z, Jiang T, *et al.* DepthVLA: enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv* 2025, arXiv:2510.13375.

[30] Pourkeshavarz M, Sigal A, Pakdamansavoji S, Li Z, Yang R, *et al.* Don't get distracted: improving robotic perception robustness via in-context visual scene editing. In *Workshop on Making Sense of Data in Robotics: composition, curation, and interpretability at scale at corl*, Munich, Germany, November 3–6, 2025.

[31] Wang S, Zhou H, Xiang D, You Y. TacRefineNet: tactile-only grasp refinement between arbitrary in-hand object poses. *arXiv* 2025, arXiv:2509.25746.

[32] Wang K, Lou S, Wang J, Yuan X. LLM based semantic fusion of infrared and visible image caption. In *2024 9th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Bangkok, Thailand, November 20–22, 2024, pp. 315–319.

[33] An T, Zhou Y, Zou H, Yang J. IoT-LLM: enhancing real-world IoT task reasoning with large language models. *arXiv* 2024 arXiv:2410.02429.

[34] Zhai X, Mustafa B, Kolesnikov A, Beyer L. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, Paris, France, October 2–6, 2023, pp. 11975–11986.

[35] Ge M, Ohtani K, Niu Y, Zhang Y, Takeda K. VLA-MP: a vision-language-action framework for multimodal perception and physics-constrained action generation in autonomous driving. *Sensors* 2025, 25(19):6163.

[36] Jain J, Yang J, Shi H. Vcoder: versatile vision encoders for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, USA, June 17–21, 2024, pp. 27992–28002.

[37] Liu R, Wang W, Yang Y. Vision-language navigation with energy-based policy, In *Proceedings of the Conference on Neural Information Processing Systems*, Vancouver, Canada, December 10–15, 2024, pp. 108208–108230.

[38] Chen Z, Wu J, Wang W, Su W, Chen G, *et al.* Internvl: scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Seattle, USA, June 17–21, 2024, pp. 24185–24198.

[39] Beyer L, Steiner A, Pinto AS, Kolesnikov A, Wang X, *et al.* Paligemma: a versatile 3B VLM for transfer. *arXiv* 2024, arXiv:2407.07726.

[40] Bai S, Chen K, Liu X, Wang J, Ge W, *et al.* Qwen2. 5-VL technical report. *arXiv* 2025, arXiv:2502.13923.

[41] Li X, Zhang H, Dong Z, Cheng X, Liu Y, *et al.* Learning fine-grained representation with token-level alignment for multimodal sentiment analysis. *Expert Syst. Appl*. 2025, 269:126274.

[42] Zhang S, Zhang X, Zhang T, Hu B, Chen Y, *et al.* Alignditil: token-level language model alignment as adaptive policy distillation, *arXiv* 2025, arXiv:2503.02832.

[43] Li C, Zhang J, Zong C. TokAlign: efficient vocabulary adaptation via token alignment. *arXiv* 2025, arXiv:2506.03523.

[44] Yan S, Han J, Tsai J, Xue H, Fang R, *et al.* Crosslmm: decoupling long video sequences from lmms via dual cross-attention mechanisms. *arXiv* 2025, arXiv:2505.17020.

[45] Jiang C, Wang Y, Yuan Q, Qu P, Li H. A 3D medical image segmentation network based on gated attention blocks and dual-scale cross-attention mechanism. *SCI Rep*. 2025, 15(1):6159.

[46] Fang L, Hou M, Huang B, Chen G, Yang J. DCAFusion: a novel general image fusion framework based on reference image reconstruction and dual-cross attention mechanism. *Inf. Sci.* 2025, 698:121772

[47] Yan F, Guo Z, Iliyasu AM, Hirota K. Multi-branch convolutional neural network with cross-attention mechanism for emotion recognition. *Sci. Rep*. 2025, 15(1):3976

[48] Wang Y, Ding P, Li L, Cui C, Ge Z, *et al.* VLA-adapter: an effective paradigm for tiny-scale vision-language-action model. *arXiv* 2025, arXiv:2509.09372.

[49] Yu Z, Wang J, Yu LC, Zhang X. Dual-encoder transformers with cross-modal alignment for multimodal aspect-based sentiment analysis. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, Taiwan, China, November 20–23, 2022, pp. 414–423.

[50] Oko K, Lin L, Cai Y, Mei S. A statistical theory of contrastive pre-training and multimodal generative ai. *arXiv* 2025, arXiv:2501.04641.

[51] Hondru V, Croitoru FA, Minaee S, Ionescu RT, Sebe N. Masked image modeling: a survey. *Int. J. Comput. Vis*. 2025, 133:1–47.

[52] Hong X, Zhang J, Li W, Lu S, Li J. Unify and Anchor: a context-aware transformer for cross-domain time series forecasting. *arXiv* 2025, arXiv:2503.01157.

[53] Jia X, You J, Zhang Z, Yan J. Drivetransformer: unified transformer for scalable end-to-end autonomous driving. *arXiv* 2025, arXiv:2503.07656.

[54] Li J, Zhu Y, Tang Z, Wen J, Zhu M, *et al.* CoA-VLA: improving vision-language-action models via visual-text chain-of-affordance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Paris, France, October 19–25, 2025, pp. 9759–9769.

[55] Zhang D, Sun J, Hu C, Wu X, Yuan Z, *et al.* Pure vision language action (VLA) models: a comprehensive survey. *arXiv* 2025, arXiv:2509.19012.

[56] Liu Z, Liu J, Xu J, Han N, Gu C, *et al.* MLA: a multisensory language-action model for multimodal understanding and forecasting in robotic manipulation. *arXiv* 2025, arXiv:2509.26642.

[57] Jang H, Yu S, Kwon H, Jeon H, Seo Y, *et al.* ContextVLA: vision-language-action model with amortized multi-frame context. *arXiv* 2025, arXiv:2510.04246.

[58] Xiong Z, Li K, Wang Z, Jackson M, Foerster J, *et al.* HyperVLA: efficient inference in vision-language-action models via hypernetworks. *arXiv* 2025, arXiv:2510.04898.

[59] Xiao E, Zhang L, Tang Y, Cheng H, Xu R, *et al.* Team Xiaomi EV-AD VLA: learning to navigate socially through proactive risk perception-technical report for IROS 2025 robosense challenge social navigation track. *arXiv* 2025, arXiv:2510.07871.

[60] Zhao H, Zhang J, Song W, Ding P, Wang D. VLA^2: empowering vision-language-action models with an agentic framework for unseen concept manipulation. *arXiv* 2025, arXiv:2510.14902.

[61] Dou Z, Zhao Q, Wan Z, Zhang D, Wang W, *et al.* Plan Then Action: high-level planning guidance reinforcement learning for LLM reasoning. *arXiv* 2025, arXiv:2510.01833.

[62] Sapkota R, Cao Y, Roumeliotis KI, Karkee M. Vision-language-action models: concepts, progress, applications and challenges. *arXiv* 2025, arXiv:2505.04769.

[63] Shek CL, Tokekar P. Option discovery using LLM-guided semantic hierarchical reinforcement learning. *arXiv* 2025, arXiv:2503.19007.

[64] Bjorck J, Castañeda F, Cherniadev N, Da X, Ding R, *et al.* GR00T N1: an open foundation model for generalist humanoid robots. *arXiv* 2025, arXiv:2503.14734.

[65] Guo W, Lu G, Deng H, Wu Z, Tang Y, *et al.* VLA-Reasoner: empowering vision-language-action models with reasoning via online monte carlo tree search. *arXiv* 2025, arXiv:2509.22643.

[66] Saxena A, Shah H, Routray S, Shah RR, Pahwa E. SITCOM: scaling inference-time compute for VLAs. *arXiv* 2025, arXiv:2510.04041.

[67] Wang Z, Li J, Zheng J, Zhang W, Liu D, *et al.* PhysiAgent: an embodied agent framework in physical world. *arXiv* 2025, arXiv:2509.24524.

[68] Karli UB, Shangguan Z, Fitzgerald T. INSIGHT: inference-time sequence introspection for generating help triggers in vision-language-action models. *arXiv* 2025, arXiv:2510.01389.

[69] Lin Z, Duan J, Fang H, Fox D, Krishna R, *et al.* FailSafe: reasoning and recovery from failures in vision-language-action models. *arXiv* 2025, arXiv:2510.01642.

[70] Wu Y, Li A, Hermans T, Ramos F, Bajcsy A, *et al.* Do what you say: steering vision-language-action models via runtime reasoning-action alignment verification. *arXiv* 2025, arXiv:2510.16281.

[71] Zhou X, Xu Y, Tie G, Chen Y, Zhang G, *et al.* LIBERO-PRO: towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv* 2025, arXiv:2510.03827.

[72] Zhao R, Ingebrand T, Chinchali S, Topcu U. MoS-VLA: a vision-language-action model with one-shot skill adaptation. *arXiv* 2025, arXiv:2510.16617.

[73] Hancock AJ, Wu X, Zha L, Russakovsky O, Majumdar A. Actions as language: fine-tuning vlms into VLAs without catastrophic forgetting. *arXiv* 2025, arXiv:2509.22195.

[74] Din MU, Akram W, Saoud LS, Rosell J, Hussain I. Vision language action models in robotic manipulation: a systematic review. *arXiv* 2025, arXiv:2507.10672.

[75] Liu J, Chen H, An P, Liu Z, Zhang R, *et al.* HybridVLA: collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv* 2025, arXiv:2503.10631.

[76] Wen J, Zhu Y, Zhu M, Tang Z, Li J, *et al.* DiffusionVLA: scaling robot foundation models via unified diffusion and autoregression. In *Proceedings of Forty-second International Conference on Machine Learning*, Vienna, Austria, July 13–19, 2025, pp. 66558–66574.

[77] Kim MJ, Finn C, Liang P. Fine-tuning vision-language-action models: optimizing speed and success. *arXiv* 2025, arXiv:2502.19645.

[78] Pertsch K, Stachowicz K, Ichter B, Driess D, Nair S, *et al.* Fast: efficient action tokenization for vision-language-action models. *arXiv* 2025, arXiv:2501.09747.

[79] Yang R, Wei H, Zhang R, Feng Z, Chen X, *et al.* Beyond human demonstrations: diffusion-based reinforcement learning to generate data for VLA training. *arXiv* 2025, arXiv:2509.19752.

[80] Chi C, Xu Z, Feng S, Cousineau E, Du Y, *et al.* Diffusion policy: visuomotor policy learning via action diffusion, *Int. J. Robot. Res*. 2025, 44(10–11):1684–1704.

[81] Wen J, Zhu M, Zhu Y, Tang Z, Li J, *et al.* Diffusion-VLA: generalizable and interpretable robot foundation model via self-generated reasoning. *arXiv* 2024, arXiv:2412.03293.

[82] Liang Z, Li Y, Yang T, Wu C, Mao S, *et al.* Discrete diffusion VLA: bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv* 2025, arXiv:2508.20072.

[83] Jiang A, Gao Y, Sun Z, Wang Y, Wang J, *et al.*DiffVLA: vision-language guided diffusion planning for autonomous driving. *arXiv* 2025, arXiv:2505.19381.

[84] Wen Y, Li H, Gu K, Zhao Y, Wang T, *et al.* LLaDA-VLA: vision language diffusion action models. *arXiv* 2025, arXiv:2509.06932.

[85] Li P, Zheng Y, Wang Y, Wang H, Zhao H, *et al.* Discrete diffusion for reflective vision-language-action models in autonomous driving. *arXiv* 2025, arXiv:2509.20109.

[86] Lyu M, Sun Y, Lin E, Li H, Chen R, *et al.* Reinforcement fine-tuning of flow-matching policies for vision-language-action models. *arXiv* 2025, arXiv:2510.09976.

[87] Zheng J, Li J, Wang Z, Liu D, Kang X, *et al.* X-VLA: soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv* 2025, arXiv:2510.10274.

[88] Li H, Ding P, Suo R, Wang Y, Ge Z, *et al.* VLA-RFT: vision-language-action reinforcement fine-tuning with verified rewards in world simulators. *arXiv* 2025, arXiv:2510.00406.

[89] Zhang H, Zhang S, Jin J, Zeng Q, Qiao Y, *et al.* Balancing signal and variance: adaptive offline RL post-training for VLA flow models. *arXiv* 2025, arXiv:2509.04063.

[90] Black K, Brown N, Driess D, Esmail A, Equi M, *et al.* A vision-language-action flow model for general robot control. *arXiv* 2024, arXiv:2410.24164.

[91] Zhang R, Dong M, Zhang Y, Heng L, Chi X, *et al.* MoLe-VLA: dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation. *arXiv* 2025, arXiv:2503.20384.

[92] Shen W, Liu Y, Wu Y, Liang Z, Gu S, *et al.* Expertise need not monopolize: action-specialized mixture of experts for vision-language-action learning. *arXiv* 2025, arXiv:2510.14300.

[93] Lu D, Gao W, Jia K. ImaginationPolicy: towards generalizable, precise and reliable end-to-end policy for robotic manipulation. *arXiv* 2025, arXiv:2509.20841.

[94] Pei X, Chen Y, Xu S, Wang Y, Shi Y, *et al.* Action-aware dynamic pruning for efficient vision-language-action manipulation. *arXiv* 2025, arXiv:2509.22093.

[95] Jang S, Kim D, Kim C, Kim Y, Shin J. Verifier-free test-time sampling for vision language action models. *arXiv* 2025, arXiv:2510.05681.

[96] Cao J, Huang Y, Guo H, Zhang R, Nan M, *et al.* Compose your policies! Improving diffusion-based or flow-based robot policies via test-time distribution-level composition. *arXiv* 2025, arXiv:2510.01068.

[97] Sendai K, Alvarez M, Matsushima T, Matsuo Y, Iwasawa Y. Leave no observation behind: real-time correction for VLA action chunks. *arXiv* 2025, arXiv:2509.23224.

[98] Wu S, Ji Y, Li Q, Zhang Z, He Q, *et al.* Dejavu: post-deployment learning for embodied agents via experience feedback. *arXiv* 2025, arXiv:2510.10181.

[99] Li C, Liu J, Wang G, Li X, Chen S, *et al.* A self-correcting vision-language-action model for fast and slow system manipulation. *arXiv* 2024, arXiv:2405.17418.

[100] O'Neill A, Rehman A, Maddukuri A, Gupta A, Padalkar A, *et al.* Open X-embodiment: robotic learning datasets and RT-X models: open X-embodiment collaboration[0]. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, May 13–17, 2024, pp. 6892–6903.

[101] Team OM, Ghosh D, Walke H, Pertsch K, Black K, *et al.* Octo: an open-source generalist robot policy. *arXiv* 2024, arXiv:2405.12213.

[102] Zheng J, Li J, Liu D, Zheng Y, Wang Z, *et al.* universal actions for enhanced embodied foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, Nashville, USA, June 15–20, 2025, pp. 22508–22519.

[103] Wen J, Zhu Y, Li J, Zhu M, Tang Z, *et al.* TinyVLA: towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robot. Autom. Lett.* 2025,10(4):3988–3996.

[104] Zhao T, Tompson J, Driess D, Florence P, Ghasemipour K, *et al.* ALOHA unleashed: a simple recipe for robot dexterity. *arXiv* 2024, arXiv:2410.13126.

[105] Belkhale S, Sadigh D.Minivla: A better VLA with a smaller footprint. The Stanford AI Lab Blog. 2024. Available: https://ai.stanford.edu/blog/minivla/ (accessed on 23 April 2026).

[106] Liu H, Li X, Li P, Liu M, Wang D, *et al.* Towards generalist robot policies: what matters in building vision-language-action models. *arXiv* 2025, arXiv:2412.14058.

[107] Zheng R, Liang Y, Huang S, Gao J, Daum éIII H, *et al.* TraceVLA: visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv* 2024, arXiv:2412.10345.

[108] Zawalski M, Chen W, Pertsch K, Mees O, Finn C, *et al.* Robotic control via embodied chain-of-thought reasoning. *arXiv* 2024, arXiv:2407.08693.

[109] Zhao Q, Lu Y, Kim MJ, Fu Z, Zhang Z, *et al.* CoT-VLA: visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, Nashville, USA, June 15–20, 2025, pp. 1702–1713.

[110] Ji Y, Tan H, Shi J, Hao X, Zhang Y, *et al.* Robobrain: a unified brain model for robotic manipulation from abstract to concrete. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, Paris, France, July 15–18, 2025, pp. 1724–1734.

[111] Zhou Z, Zhu Y, Zhu M, Wen J, Liu N, *et al.* ChatVLA: unified multimodal understanding and robot control with vision-language-action model. *arXiv* 2025, arXiv:2502.14420.

[112] Ye S, Jang J, Jeon B, Joo S, Yang J, *et al.* Latent action pretraining from videos, *arXiv* 2024, arXiv:2410.11758.

[113] Schmidt D, Jiang M. Learning to act without actions. *arXiv* 2023, arXiv:2312.10812.

[114] Lipman Y, Chen RT, Ben-Hamu H, Nickel M, Le M. Flow matching for generative modeling. *arXiv* 2022, arXiv:2210.02747.

[115] Liu Q. Rectified flow: a marginal preserving approach to optimal transport. *arXiv* 2022, arXiv:2209.14577.
[116] Liu S, Wu L, Li B, Tan H, Chen H, *et al.* RDT-1B: a diffusion foundation model for bimanual manipulation. *arXiv* 2024, arXiv:2410.07864.
[117] Li Q, Liang Y, Wang Z, Luo L, Chen X, *et al.* CogACT: a foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv* 2024, arXiv:2411.19650.
[118] Yan F, Liu F, Zheng L, Zhong Y, Huang Y, *et al.* RoboTron-Mani: all-in-one multimodal large model for robotic manipulation. *arXiv* 2024, arXiv:2412.07215.
[119] Hou Z, Zhang T, Xiong Y, Duan H, Pu H, *et al.* Dita: scaling diffusion transformer for generalist vision-language-action policy. *arXiv* 2025, arXiv:2503.19757.
[120] Goyal A, Hadfield H, Yang X, Blukis V, Ramos F. VLA-0: building state-of-the-art VLAs with zero modification. *arXiv* 2025, arXiv:2510.13054.
[121] Reuss M, Yağmurlu ÖE, Wenzel F, Lioutikov R. Multimodal diffusion transformer: learning versatile behavior from multimodal goals. *arXiv* 2024, arXiv:2407.05996.
[122] Tian Y, Yang S, Zeng J, Wang P, Lin D, *et al.* Predictive inverse dynamics models are scalable learners for robotic manipulation. *arXiv* 2024, arXiv:2412.15109.
[123] Zhang Z, Li H, Dai Y, Zhu Z, Zhou L, *et al.* From spatial to actions: grounding vision-language-action model in spatial foundation priors. *arXiv* 2025, arXiv:2510.17439.
[124] Dong Z, Wang X, Zhu Z, Wang Y, Wang Y, *et al.* EMMA: generalizing real-world robot manipulation via generative visual transfer. *arXiv* 2025, arXiv:2509.22407.
[125] Bu Q, Yang Y, Cai J, Gao S, Ren G, *et al.* UniVLA: learning to act anywhere with task-centric latent actions. *arXiv* 2025, arXiv:2505.06111.
[126] Chen X, Guo J, He T, Zhang C, Zhang P, *et al.* IGOR: image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv* 2024, arXiv:2411.00785.
[127] Hu Y, Guo Y, Wang P, Chen X, Wang Y, *et al.* Video prediction policy: a generalist robot policy with predictive visual representations. *arXiv* 2024, arXiv:2412.14803.
[128] Wei Z, Xu Z, Guo J, Hou Y, Gao C, *et al. D(R, O)* Grasp: a unified representation of robot and object interaction for cross-embodiment dexterous grasping. *arXiv* 2024, arXiv:2410.01702.
[129] Gao C, Jiang Y, Chen F. Transferring hierarchical structures with dual meta imitation learning. In *Proceedings of the Conference on Robot Learning*, Atlanta, USA, November 6–9, 2023, pp. 762–773.
[130] Intelligence P, Black K, Brown N, Darpinian J, Dhabalia K, *et al.* $\pi 0.5$: a vision-language-action model with open-world generalization. *arXiv* 2025, arXiv:2504.16054.
[131] Qwen AY, Yang B, Zhang B, Hui B, Zheng B, *et al.* Qwen2 technical report. *arXiv* 2024, arXiv:2407.10671.
[132] Team G, Georgiev P, Lei VI, Burnell R, Bai L, *et al.* Gemini 1.5: unlocking multimodal understanding across millions of tokens of context. *arXiv* 2024, arXiv:2403.05530.
[133] Gao C, Li Z, Gao H, Chen F. Iterative interactive modeling for knotting plastic bags. In *Proceedings of the Conference on Robot Learning*, Atlanta, USA, November 6–9, 2023, pp. 571–582.
[134] Chen H, Li J, Wu R, Liu Y, Hou Y, *et al.* Metafold: language-guided multi-category garment folding framework via trajectory generation and foundation model. *arXiv* 2025, arXiv:2503.08372.
[135] Belkhale S, Ding T, Xiao T, Sermanet P, Vuong Q, *et al.* RT-H: action hierarchies using language. *arXiv* 2024, arXiv:2403.01823.

[136] Shi L X, Ichter B, Equi M, Ke L, Pertsch K, *et al.* Hi robot: open-ended instruction following with hierarchical vision-language-action models, *arXiv* 2025, arXiv:2502.19417.

[137] Li Y, Deng Y, Zhang J, Jang J, Memmel M, *et al.* Hamster: hierarchical action models for open-world robot manipulation. *arXiv* 2025, arXiv:2502.05485.

[138] Lin T, Li G, Zhong Y, Zou Y, Du Y, *et al.* Evo-0: vision-language-action model with implicit spatial understanding. *arXiv* 2025, arXiv:2507.00416.

[139] Nasiriany S, Kirmani S, Ding T, Smith L, Zhu Y, *et al.* RT-Affordance: affordances are versatile intermediate representations for robot manipulation. In *Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, USA, May 19–23, 2025, pp. 8249–8257.

[140] Mendonca R, Bahl S, Pathak D. Structured world models from human videos. *arXiv* 2023, arXiv:2308.10901.

[141] Yuan W, Duan J, Blukis V, Pumacay W, Krishna R, *et al.* Robopoint: a vision-language model for spatial affordance prediction for robotics. *arXiv* 2024, arXiv:2406.10721.

[142] Gao C, Zhang H, Xu Z, Cai Z, Shao L. Flip: flow-centric generative planning as general-purpose manipulation world model. *arXiv* 2024, arXiv:2412.08261.

[143] Wen J, Zhu Y, Li J, Tang Z, Shen C, *et al.* DexVLA: vision-language model with plug-in diffusion expert for general robot control. *arXiv* 2025, arXiv:2502.05855.

[144] Xu Z, Gao C, Liu Z, Yang G, Tie C, *et al.* Manifoundation model for general-purpose robotic manipulation of contact synthesis with arbitrary objects and robots. In *Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Abu Dhabi, United Arab Emirates, October 14–18, 2024, pp. 10905–10912.

[145] Lin F, Nai R, Hu Y, You J, Zhao J, *et al.* OneTwoVLA: a unified vision-language-action model with adaptive reasoning. *arXiv* 2025, arXiv:2505.11917.

[146] Huang Z, Liu M, Lin X, Zhu M, Zhao C, *et al.* NotVLA: narrowing of dense action trajectories for generalizable robot manipulation. *arXiv* 2025, arXiv:2510.03895.

[147] Abdolmaleki A, Abeyruwan S, Ainslie J, Alayrac JB, Arenas MG, *et al.* Gemini robotics 1.5: pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv* 2025, arXiv:2510.03342.

[148] Gu J, Kirmani S, Wohlhart P, Lu Y, Arenas MG, *et al.* RT-Trajectory: robotic task generalization via hindsight trajectory sketches. *arXiv* 2023, arXiv:2311.01977.

[149] Du Y, Yang S, Dai B, Dai H, Nachum O, *et al.* Learning universal policies via text-guided video generation, *Adv. Neural Inf. Process. Syst*. 2023, 36:9156–9172.

[150] Yang M, Du Y, Ghasemipour K, Tompson J, Schuurmans D, *et al.* Learning interactive real-world simulators. *arXiv* 2023, arXiv:2310.06114.

[151] Chen Y, Chen Z, Yin J, Huo J, Tian P, *et al.* Gravmad: grounded spatial value maps guided action diffusion for generalized 3d manipulation. *arXiv* 2024, arXiv:2409.20154.

[152] Chen Y, Chen Z, Chan NT, Chen J, Yin J, *et al.* RoboHiMan: a hierarchical evaluation paradigm for compositional generalization in long-horizon manipulation. *arXiv* 2025, arXiv:2510.13149.

[153] Wang Z, Zhou Z, Song J, Huang Y, Shu Z, *et al.* VLATest: testing and evaluating vision-language-action models for robotic manipulation. *Proc. ACM Softw. Eng*. 2025, 2(FSE):1615–1638.

[154] Zhao W, Li G, Gong Z, Ding P, Zhao H, *et al.* Unveiling the potential of vision-language-action models with open-ended multimodal instructions. *arXiv* 2025, arXiv:2505.11214.

[155] Zhang Z, Xu H, Yang Z, Yue C, Lin Z, *et al.* TA-VLA: elucidating the design space of torque-aware vision-language-action models. *arXiv* 2025, arXiv:2509.07962.

[156] Zhao W, Ding P, Zhang M, Gong Z, Bai S, *et al.* VLAs: vision-language-action model with speech instructions for customized robot manipulation. *arXiv* 2025, arXiv:2502.13508.

[157] Li H, Chen Y, Cui W, Liu W, Liu K, *et al.* Survey of vision-language-action models for embodied manipulation. *arXiv* 2025, arXiv:2508.15201.