

Review

Not peer-reviewed version

---

# Vision-Language-Action (VLA) Models for Unmanned Aerial Robotics and Bimanual Manipulation: A Review

---

[Inkyu Sa](#)<sup>\*</sup>, Chanoh Park, [Ho Seok Ahn](#)

Posted Date: 9 April 2026

doi: 10.20944/preprints202604.0664.v1

Keywords: Vision-Language-Action models; bimanual manipulation; unmanned aerial robotics; drones; UAV; unmanned systems; robot learning; imitation learning; flow matching



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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

Review

# Vision-Language-Action (VLA) Models for Unmanned Aerial Robotics and Bimanual Manipulation: A Review

Inkyu Sa <sup>1,\*</sup>, Chanoh Park <sup>2</sup> and Ho Seok Ahn <sup>3</sup>

<sup>1</sup> Chef Robotics

<sup>2</sup> RovifyLab

<sup>3</sup> University of Auckland

\* Correspondence: inkyu@chefrobotics.ai

## Abstract

Vision-Language-Action (VLA) models unify visual perception, natural-language understanding, and action generation within a single foundation model, allowing a robot to follow instructions such as “fold the towel” or “fly to the red building” directly from camera images. Because VLAs inherit world knowledge from internet-scale pre-training, they have become the dominant framework for learning-based manipulation, with bimanual coordination serving as the most demanding testbed: two arms with 7+ degrees of freedom each must move in concert to fold, assemble, and reorient objects. Unmanned aerial robotics faces a structurally similar challenge: a drone must coordinate thrust, attitude, and increasingly gripper commands from visual observations under strict latency and payload constraints. This review covers 186 contributions spanning 2017–2026 and organized along seven dimensions: VLA architectures, training recipes, action representations, bimanual coordination (2022–2026), unmanned aerial vehicle (UAV) navigation and control (2017–2026), language grounding, and cross-cutting concerns including memory and world models. We show that the coordination strategies, training recipes, and action representations developed for bimanual VLAs transfer to unmanned aerial systems, and identify fourteen research directions across both domains.

**Keywords:** Vision-Language-Action models; bimanual manipulation; unmanned aerial robotics; drones; UAV; unmanned systems; robot learning; imitation learning; flow matching

## 1. Introduction

*Vision-Language-Action* (VLA) models use a single foundation model to map camera images and language instructions to robot actions. A VLA processes visual and language inputs through a Vision-Language Model (VLM) pre-trained on internet-scale data, then generates motor commands through a learned action head. Because the architecture makes no assumptions about the specific robot, VLAs can control manipulators, mobile robots, and drones with the same model family, enabling robots to assist in homes, factories, and disaster-response scenarios.

To date, the vast majority of VLA research has focused on manipulation, and bimanual coordination in particular. Bimanual tasks (folding laundry, assembling boxes, clearing tables) require two 7+-degree-of-freedom arms to move in concert under partial observability, making them among the most challenging testbeds for VLA models. This concentration of research effort means that bimanual manipulation is where VLA architectures, training recipes, and action representations are best understood. We therefore devote the first application section of this review to a detailed analysis of VLAs for bimanual manipulation.

We then extend the analysis to unmanned aerial robotics, where the same VLA ideas are now being adopted. The connection between the two domains is not merely conceptual. Coordinating two arms and coordinating a drone fleet both require generating coupled multi-agent actions from shared

observations. The action chunking methods that produce smooth bimanual trajectories also produce smooth flight paths. Drones with grippers or robotic arms face both challenges at once, stabilizing flight while manipulating objects. The training recipes (pre-training on diverse data, sim-to-real transfer, reinforcement learning from practice) are shared. Language grounding is also unified: the same VLM mechanisms that interpret “fold the shirt neatly” for a manipulator interpret “fly to the red building and inspect the roof” for a drone. Reviewing bimanual VLAs first provides the vocabulary and analytical framework that makes the aerial discussion concrete.

**Progress in manipulation VLAs.** The field has moved fast. **RT-2** [1] (2023) first showed that a VLM can be fine-tuned to output robot actions.  $\pi_0$  [2] (2024) introduced flow matching, a method that learns to transform random noise into robot actions. It reached state-of-the-art bimanual performance on tasks like laundry folding and box assembly.  $\pi_{0.5}$  [3] (2025) deployed VLAs in real homes with a 98% success rate, and  $\pi_0^*$  [4] enabled VLAs to improve from their own practice via reinforcement learning. Open-source systems (**OpenVLA** [5], **Octo** [6]) and efficient architectures [7,8] have made the technology broadly accessible.

**Emergence of unmanned aerial VLAs.** In parallel, the unmanned aerial systems community has begun adopting VLA ideas. **CognitiveDrone** [9] generates real-time flight commands from camera images and text instructions. **DroneVLA** [10] and **AIR-VLA** [11] perform language-commanded aerial grasping. **Flying Hand** [12] uses the same action chunking method developed for bimanual manipulation (ACT) on a hexarotor with a robotic arm. These systems confirm that the VLA framework transfers across embodiments.

**Gap in existing surveys.** Surveys on foundation models for robotics [13] address high-level planning but not low-level motor control. Reviews of imitation learning [14] predate VLAs. Surveys on multi-arm systems [15,16] cover classical methods, not learned policies. Aerial surveys have examined perception and detection but not end-to-end VLA-based drone control. No existing review treats bimanual manipulation and unmanned aerial robotics as two instances of the same VLA problem.

This review fills that gap by treating VLAs as a single framework applied to two embodiment families. We first review the shared VLA machinery (architectures, training recipes, action representations, language grounding) and then apply it to bimanual manipulation and unmanned aerial robotics in turn, drawing explicit parallels throughout. The main contributions are:

- A unified taxonomy of VLA models covering architectures, training, action representations, bimanual manipulation, and unmanned aerial robotics, with comparison tables spanning 30+ methods.
- The first cross-domain analysis connecting bimanual coordination strategies to multi-drone and aerial manipulation systems, showing how insights transfer between embodiments.
- Fourteen research directions identifying open challenges across both domains, from real-time control and safety certification to end-to-end drone VLAs and bridging the research-to-production gap.

The paper is structured to build from shared foundations to domain-specific applications. Sections 2–7 cover the common VLA stack: problem formulation, background, benchmarks, architectures, training, and action representations. Section 8 then applies this stack to bimanual manipulation, where VLAs are most mature. Section 9 applies it to unmanned aerial robotics, drawing on the bimanual analysis to highlight what transfers and what differs. Section 10 examines language grounding across both domains. Section 11 addresses cross-cutting concerns (visual representations, world models, memory, safety, sim-to-real). Section 12 synthesizes findings and identifies research directions that span both embodiment families.

## 2. Problem Definition and Scope

We begin by formalizing the core concepts that underpin the review: the VLA policy, action chunking, flow matching for action generation, and bimanual coordination. The notation introduced

here is used consistently in subsequent sections; Table 1 provides a summary. Figure 1 presents the taxonomy that organizes this review.

**Table 1.** Summary of notation used in this review.

| Symbol   | Description   |
|--|---|
| $\pi_\theta$                                     | VLA policy parameterized by $\theta$                                  |
| $\mathbf{o}_t$                                   | Visual observation at time $t$  |
| $\ell$   | Natural-language instruction  |
| $\mathbf{q}_t$                                   | Proprioceptive state (joint positions)                                |
| $\mathbf{a}_t$                                   | Single-step action  |
| $\mathbf{A}_t$                                   | Action chunk of horizon $H$   |
| $H$  | Action chunk horizon (number of steps)                                |
| $d_a$  | Action dimensionality   |
| $K$  | Number of denoising/flow steps  |
| $\mathbf{v}_\theta$                              | Learned velocity field (flow matching)                                |
| $\mathbf{a}_t^L, \mathbf{a}_t^R$                 | Left and right arm actions  |
| $f_{\text{vis}}, f_{\text{VLM}}, f_{\text{act}}$ | Visual encoder, VLM backbone, action head                             |
| $\alpha, \gamma, \sigma_k$                       | Diffusion schedule coefficients (Equation 9)                          |
| $\mathbf{z}$                                     | Gaussian noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ |
| $\epsilon_\theta$                                | Noise prediction network (diffusion)                                  |

### 2.1. VLA Policy Formulation

A *Vision-Language-Action* model defines a policy  $\pi_\theta$  parameterized by  $\theta$  that maps a visual observation  $\mathbf{o}_t \in \mathcal{O}$ , a language instruction  $\ell \in \mathcal{L}$ , and optionally a proprioceptive state  $\mathbf{q}_t \in \mathcal{Q}$  to an action  $\mathbf{a}_t \in \mathcal{A}$ :

$$\pi_\theta : \mathcal{O} \times \mathcal{L} \times \mathcal{Q} \rightarrow \mathcal{A}. \quad (1)$$

The observation space  $\mathcal{O}$  typically consists of one or more camera images  $\mathbf{I}_t \in \mathbb{R}^{H_{\text{img}} \times W_{\text{img}} \times 3}$ . The language instruction  $\ell$  is a natural-language string tokenized and embedded by the VLM backbone. The action space  $\mathcal{A}$  varies by embodiment; for a single  $n$ -DOF arm with a gripper,  $\mathbf{a}_t \in \mathbb{R}^{n+1}$ , encoding either joint velocities or end-effector displacements plus a gripper command.

The VLA framework distinguishes itself from prior vision-based control policies by sharing a backbone with a pre-trained VLM. Concretely, a VLA typically consists of three components: (1) a visual encoder  $f_{\text{vis}}$  that produces image tokens, (2) a vision-language backbone  $f_{\text{VLM}}$  that jointly reasons over image and text tokens, and (3) an action head  $f_{\text{act}}$  that decodes actions from the VLM's hidden representations:

$$\mathbf{a}_t = f_{\text{act}}(f_{\text{VLM}}(f_{\text{vis}}(\mathbf{I}_t), \text{Tok}(\ell), \mathbf{q}_t)), \quad (2)$$

where  $\text{Tok}(\ell)$  denotes the tokenized language instruction. The proprioceptive state  $\mathbf{q}_t$  is likewise tokenized and fed into  $f_{\text{VLM}}$  alongside the visual and language tokens.

### 2.2. Action Chunking

Rather than predicting a single action  $\mathbf{a}_t$ , modern VLA policies predict an *action chunk*, a sequence of  $H$  future actions, in a single forward pass:

$$\mathbf{A}_t = (\mathbf{a}_t, \mathbf{a}_{t+1}, \dots, \mathbf{a}_{t+H-1}) \in \mathbb{R}^{H \times d_a}, \quad (3)$$

where  $H$  is the *chunk horizon* and  $d_a$  is the action dimension. Action chunking, introduced in the context of **ACT** [17], offers two key advantages. First, it amortizes the cost of a single VLM forward pass over multiple control steps, which allows high-frequency control despite the latency of large models. Second, it captures temporal correlations between successive actions, producing smoother trajectories than single-step prediction. The chunk is typically executed open-loop or with temporal ensembling, where overlapping chunks are averaged to reduce jitter.

### 2.3. Flow Matching for Action Generation

Flow matching [18] provides a framework for learning continuous normalizing flows by regressing a vector field that transports samples from a simple prior  $p_0$  (e.g., a standard Gaussian) to the data distribution  $p_1$ . Given a time-dependent vector field  $\mathbf{v}_\theta(\mathbf{x}, t)$  for  $t \in [0, 1]$ , the flow  $\phi_t(\mathbf{x})$  satisfies (here  $t$  denotes the continuous flow time parameter, distinct from the discrete control step index used elsewhere):

$$\frac{d}{dt}\phi_t(\mathbf{x}) = \mathbf{v}_\theta(\phi_t(\mathbf{x}), t), \quad \phi_0(\mathbf{x}) = \mathbf{x}, \quad \mathbf{x} \sim p_0. \quad (4)$$

where  $\phi_t$  is the flow map at time  $t$ , transporting a sample from  $p_0$  toward  $p_1$ . The training objective minimizes the conditional flow matching loss:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t, \mathbf{x}_0, \mathbf{x}_1} \left[ \|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\mathbf{x}_1 - \mathbf{x}_0)\|^2 \right], \quad (5)$$

where  $\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\mathbf{x}_1$  is a linear interpolation. In the VLA context,  $\mathbf{x}_1$  is the ground-truth action chunk  $\mathbf{A}_t$  and  $\mathbf{x}_0$  is Gaussian noise.  $\pi_0$  [2] applies this formulation with a VLM backbone: the VLM hidden states condition the flow, and the action head iteratively denoises a noisy action chunk over  $K$  steps during inference.

### 2.4. Bimanual Coordination

For a bimanual system with a left arm and a right arm, the joint action space is:

$$\mathbf{a}_t^{\text{bi}} = [\mathbf{a}_t^L; \mathbf{a}_t^R] \in \mathbb{R}^{d_L + d_R}, \quad (6)$$

where  $\mathbf{a}_t^L \in \mathbb{R}^{d_L}$  and  $\mathbf{a}_t^R \in \mathbb{R}^{d_R}$  are the actions for the left and right arms respectively. For typical 7-DOF arms with grippers,  $d_L = d_R = 8$  (7 joint positions or velocities + 1 gripper command), yielding  $d_a = 16$ . With action chunking of horizon  $H$ , the full bimanual action chunk has dimensionality  $H \times (d_L + d_R)$ , which for typical settings ( $H = 50$ ,  $d_a = 16$ ) reaches 800 dimensions.

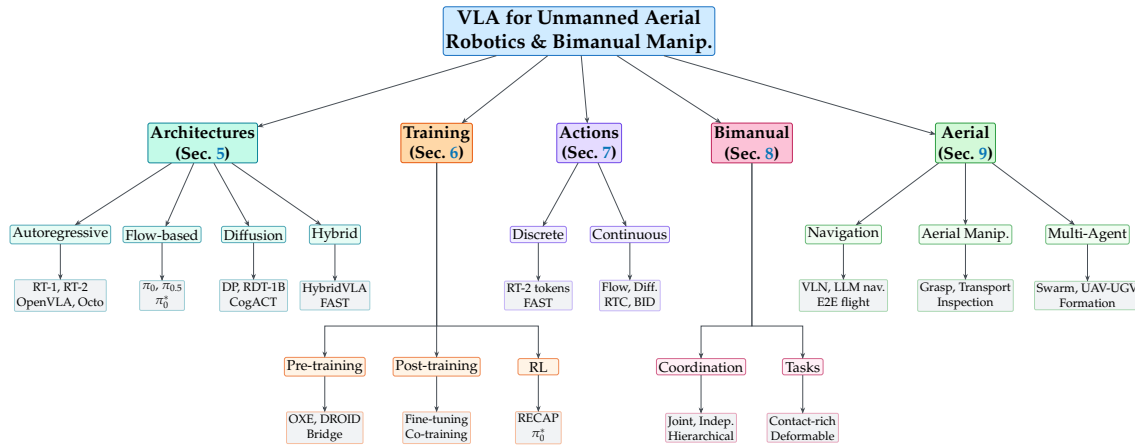
Bimanual coordination can be categorized into three modes:

1. **Independent:** Each arm executes its own subtask without coupling (e.g., one arm picks an object while the other holds a container).
2. **Loosely coupled:** Arms must coordinate timing but not forces (e.g., handover tasks where one arm releases as the other grasps).
3. **Tightly coupled:** Arms must coordinate both motion and forces simultaneously (e.g., folding fabric where both arms must apply tension).

### 2.5. Scope of This Review

This review covers VLA models that integrate a pre-trained vision-language backbone with an action generation mechanism, with emphasis on their application to bimanual manipulation and unmanned aerial robotics. We include autoregressive, flow-based, diffusion-based, and hybrid architectures published through early 2026. We focus on learning-based approaches trained from demonstrations or reinforcement learning; classical motion planning, optimization-based bimanual coordination, and traditional PID-based drone controllers are outside our scope. For bimanual motion planning, we refer readers to Abbas *et al.* [15]; for classical aerial control, we refer to standard flight dynamics references.

With these definitions established, Section 3 reviews the prerequisite concepts.



**Figure 1.** Taxonomy of VLA models for bimanual manipulation and unmanned aerial robotics. The review is organized along five major dimensions: architectural foundations (autoregressive, flow-based, diffusion-based, hybrid), training recipes (pre-training, post-training, reinforcement learning), action representations (discrete tokenization, continuous generation), bimanual-specific concerns (coordination strategies, task types), and unmanned aerial robotics (navigation, aerial manipulation, multi-agent unmanned systems). Each branch is covered in a dedicated section.

### 3. Background

Before surveying specific VLA methods, we review the foundational concepts they build upon: vision-language models, imitation learning, generative modeling for action generation, bimanual robotic systems, and aerial robotic systems.

#### 3.1. Vision-Language Models

Vision-Language Models (VLMs) jointly process visual and textual inputs, built upon the Transformer architecture [19] and trained on internet-scale image-text datasets. The Vision Transformer (ViT) [20] extended self-attention to image patches, while CLIP [21] established contrastive pre-training for aligned visual-textual representations. The pre-train-then-fine-tune recipe, scaled by GPT-3 [22] and refined via instruction-tuning [23], is the foundation-model methodology that VLAs inherit.

Key VLMs relevant to this review include: **PaLM-E** [24], which demonstrated embodied reasoning in a 562B-parameter model; **PaLIGemma** [25] and **Gemma** [26], which provide efficient open-weight backbones used by several VLA systems; and open-source models (**LLaMA** [27], **LLaVA** [28]) that democratized access. VLMs are attractive for robotics because they recognize objects, understand spatial relationships, and interpret instructions without robotics-specific training.

The transition from VLM to VLA requires adding an action output modality. This can be achieved by (1) tokenizing actions as text tokens and fine-tuning the VLM’s language head [1], (2) attaching a separate action head that reads from the VLM’s hidden states [2], or (3) using the VLM as a high-level planner that conditions a low-level policy [29]. Each approach trades off between exploiting pre-trained knowledge and accommodating the continuous, high-frequency nature of robot control. A limitation for robotics is that VLMs lack grounding in physical interaction dynamics; they recognize objects but cannot predict contact forces or material deformation.

#### 3.2. Imitation Learning

Imitation learning (IL) trains a policy  $\pi_\theta$  to mimic expert demonstrations  $\mathcal{D} = \{(\mathbf{o}_i, \ell_i, \mathbf{a}_i)\}_{i=1}^N$ , dating back to ALVINN [30]. The simplest form, *behavioral cloning* (BC), minimizes a supervised loss:

$$\mathcal{L}_{\text{BC}} = \mathbb{E}_{(\mathbf{o}, \ell, \mathbf{a}) \sim \mathcal{D}} \left[ \|\pi_\theta(\mathbf{o}, \ell) - \mathbf{a}\|^2 \right]. \quad (7)$$

BC suffers from compounding errors due to distribution shift [31]: at test time the policy visits states not seen in training. Action chunking [17] mitigates this by reducing decision points. A second challenge is *multimodality*: for a given observation, multiple valid action sequences may exist. Mean-squared-error regression averages over modes, producing invalid intermediate actions. Bimanual tasks amplify both problems because the state space is higher-dimensional and errors propagate across both arms. This motivates expressive generative models (diffusion, flow matching, autoregressive sampling) as action decoders. **Language-conditioned IL** [32] extends BC by conditioning on language instructions; VLAs take this further by using pre-trained VLMs for rich semantic grounding.

### 3.3. Generative Modeling for Actions

Three families of generative models underpin VLA action generation. Early approaches used VAEs [33] and GANs [34] for latent action representations. DDPMs [35] and score-based models [36] provided higher-fidelity generation at the cost of iterative sampling, with **Latent Diffusion Models** [37] reducing this cost via learned latent spaces. The **Decision Transformer** [38] reframed RL as sequence modeling, foreshadowing the autoregressive approach that VLAs later adopted, and **Gato** [39] extended this to a generalist agent handling text, images, and robotic actions in one Transformer.

#### 3.3.1. Autoregressive Models

Autoregressive models factorize the action distribution as a product of conditionals:

$$p(\mathbf{A}_t | \mathbf{o}_t, \ell) = \prod_{h=0}^{H-1} p(\mathbf{a}_{t+h} | \mathbf{a}_{t:t+h-1}, \mathbf{o}_t, \ell). \quad (8)$$

**RT-2** [1] discretizes continuous actions into 256 bins per dimension and generates action tokens left-to-right using the VLM’s language modeling head. This approach naturally exploits VLM pre-training but introduces quantization error and sequential latency that scales with action dimensionality.

#### 3.3.2. Diffusion Models

**Diffusion Policy** [40] generates actions by iteratively denoising a Gaussian sample through a learned reverse diffusion process:

$$\mathbf{A}_t^{(k-1)} = \alpha \left( \mathbf{A}_t^{(k)} - \gamma \epsilon_\theta(\mathbf{A}_t^{(k)}, k, \mathbf{o}_t) \right) + \sigma_k \mathbf{z}, \quad (9)$$

where  $\epsilon_\theta$  is the noise prediction network,  $k$  indexes the denoising step,  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  is standard Gaussian noise, and  $\alpha, \gamma, \sigma_k$  are schedule-dependent coefficients. Diffusion models excel at capturing distributions over multiple valid actions and produce smooth trajectories, but require multiple denoising steps ( $K = 10\text{--}100$ ), increasing inference latency.

#### 3.3.3. Flow Matching

Flow matching [18], formalized in Equation 5, offers a simpler training objective and often requires fewer steps than diffusion. **Rectified Flow** [41] straightens transport paths to reduce integration steps.  $\pi_0$  [2] demonstrated that flow matching with  $K = 10$  steps produces high-quality action chunks at 50 Hz for bimanual systems.

### 3.4. Bimanual Robotic Systems

Three hardware platforms have transformed bimanual VLA research (see Table 8 for specifications). **ALOHA** [17] provides low-cost bilateral teleoperation for two 6-DOF arms, paired with the ACT policy (Action Chunking with Transformers) that predicts action chunks at 50 Hz. **Mobile ALOHA** [42] extends this to a mobile base and demonstrated co-training (mixing target-task data with diverse data), which directly influenced VLA training recipes (Section 6.2). **UMI** [43] decouples data collection from the robot via hand-held grippers with visual-inertial tracking, allowing demonstrations in diverse

environments without teleoperation hardware. The standardization of action spaces across these platforms has facilitated cross-system transfer; data collection strategies are detailed in Section 6.4.

Two practical concerns affect bimanual VLA deployment. *Calibration*: even small errors ( $\sim 1$  cm position,  $\sim 2^\circ$  orientation) between arms can cause policies to fail; **UMI** [43] addresses this via visual-inertial tracking that decouples data collection from arm calibrations. *Action space choice*: joint-space actions (**ACT** [17], **RDT-1B** [44]) provide direct control but are embodiment-specific, while end-effector actions ( $\pi_0$  [2]) facilitate cross-embodiment transfer at the cost of inverse kinematics errors.

### 3.5. Aerial Robotic Systems

A quadrotor is a 6-DOF rigid body (3 translational, 3 rotational) controlled through differential thrust of four rotors, making it underactuated (4 inputs for 6 DOF). This underactuation creates inherent coupling between translational and rotational motion that complicates learned control policies. Quadrotors are the dominant platform for learning-based unmanned aerial robotics due to their mechanical simplicity, hovering capability, and commercial availability.

Traditional drone control employs cascaded PID loops operating at  $\geq 250$  Hz, with an inner attitude loop and an outer position loop. Learning-based approaches replace part or all of this pipeline with neural network policies. The action space varies from high-level waypoints (suitable for navigation VLAs operating at 5–10 Hz) to low-level motor commands (required for agile flight at  $\geq 100$  Hz). This range of control frequencies and abstraction levels parallels the hierarchy observed in manipulation VLAs, from high-level subgoal generation ( $\pi_{0.5}$ ) to low-level continuous action chunks ( $\pi_0$ ).

High-fidelity simulators play an outsized role in aerial VLA development. **AirSim** [45] provides photorealistic rendering via Unreal Engine with accurate quadrotor dynamics. **Flightmare** [46] decouples rendering from physics, allowing massively parallel RL training at  $200\times$  real-time. These simulators are to aerial VLAs what **LIBERO** and **SIMPLER** are to manipulation VLAs: essential infrastructure for training and evaluation at scale.

The datasets and benchmarks that drive VLA development and evaluation are reviewed in the next section.

## 4. Datasets, Benchmarks, and Evaluation

Large-scale datasets and standardized benchmarks form the infrastructure that drives VLA research. We review the major datasets used for pre-training and evaluation, simulation benchmarks, and the metrics employed to assess bimanual manipulation and aerial navigation performance.

### 4.1. Pre-training Datasets

VLA training relies on large-scale robot demonstration data for pre-training. Table 2 compares the major datasets. Three have proved most influential.

**Open X-Embodiment (OXE)** [47] is the largest open robot dataset, aggregating demonstrations from over 20 institutions across 22 robot embodiments. It contains more than 1 million episodes spanning single-arm, bimanual, and mobile manipulation tasks. OXE’s diversity in embodiments, viewpoints, and environments makes it the standard pre-training corpus for cross-embodiment VLAs. **OpenVLA** [5], **Octo** [6], and  $\pi_0$  [2] all use OXE (or subsets thereof) for pre-training.

**DROID** [48] provides 76,000 trajectories collected across 564 scenes and 86 tasks using Franka Emika arms. Unlike OXE, DROID emphasizes diversity within a single embodiment: 50 operators collected data across varied environments, capturing natural scene diversity. DROID has been shown to improve generalization when included in VLA pre-training mixtures.

**BridgeData V2** [49], building on the original BridgeData [50] that first demonstrated cross-domain dataset boosting, contains 60,096 trajectories from a WidowX robot performing tabletop manipulation tasks across 24 environments. Its relatively uniform setup and reliable labeling make it a standard evaluation dataset. Many VLA papers report results on Bridge tasks.

**GigaBrain-0.5M** [51] is a recent large-scale dataset containing 500,000 episodes collected via a combination of teleoperation and autonomous data collection. It includes bimanual manipulation episodes and was designed to support VLA training with world-model-based reinforcement learning.

#### 4.2. Simulation Benchmarks

Simulation benchmarks support reproducible evaluation at scale without the expense and variability of real-world experiments.

**LIBERO** [52] is a benchmark suite of 130 language-conditioned manipulation tasks organized into five suites: LIBERO-Spatial (spatial relationship understanding), LIBERO-Object (novel object generalization), LIBERO-Goal (goal specification comprehension), LIBERO-Long (multi-step task execution), and LIBERO-100 (a larger training set). The four evaluation suites each contain 10 tasks with 50 demonstrations per task. LIBERO has become the primary simulation benchmark for VLA evaluation because it tests multiple generalization axes independently, allowing researchers to diagnose specific weaknesses.

**SIMPLER** [53] provides simulated counterparts to real-world evaluation setups, which allows VLA evaluation without physical hardware. It includes tasks from the Bridge and Google Robot environments, with visual fidelity and physics parameters calibrated to correlate with real-world performance. SIMPLER’s key contribution is demonstrating that simulation success rates predict real-world success rates with  $r > 0.8$  correlation for most task categories. This validates simulation as a low-cost proxy for real evaluation.

Other simulation platforms include **RLBench** [54] (100 procedurally generated tasks), **MetaWorld** [55] (50 parametric tasks), **RoboSuite** [56], **ManiSkill2** [57] (soft-body tasks), and **BEHAVIOR-1K** [58] (1,000 household activities). Li *et al.* [59] found that both physics fidelity and visual realism matter for reliable simulation-to-real prediction.

Neither LIBERO nor SIMPLER includes bimanual tasks, a significant limitation. Bimanual simulation requires dual-arm physics and contact-rich interaction modeling that no standard benchmark provides.

#### 4.3. Evaluation Metrics

VLA evaluation relies on several complementary metrics:

**Task success rate** is the primary metric, defined as the fraction of  $N$  evaluation episodes in which the robot completes the specified task:

$$SR = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[\text{task}_i \text{ completed}], \quad (10)$$

where  $\mathbb{I}[\cdot]$  is the indicator function.

**Normalized score** averages success rates across multiple task suites, which allows comparison across benchmarks with different numbers of tasks.

**Inference latency** measures the wall-clock time for a single action chunk prediction, critical for real-time control. In dual-arm setups running at 50 Hz, the action generation must complete within 20 ms per step (or within  $H \times 20$  ms per chunk).

**Data efficiency** tracks how many demonstrations are required to reach a target success rate, relevant for bimanual tasks where data collection is expensive.

**Table 2.** Comparison of major robot datasets used for VLA pre-training and evaluation. Bimanual coverage indicates whether the dataset includes bimanual manipulation episodes.

| Dataset                    | Episodes | Embodiments | Tasks | Bimanual | Language | Year |
|----------------------------|----------|-------------|-------|----------|----------|------|
| <b>OXE</b> [47]            | >1M      | 22          | 527   | ✓        | ✓        | 2024 |
| <b>DROID</b> [48]          | 76K      | 1 (Franka)  | 86    | –        | ✓        | 2024 |
| <b>BridgeData V2</b> [49]  | 60K      | 1 (WidowX)  | 13    | –        | ✓        | 2023 |
| <b>ALOHA</b> [17]          | ~1K      | 1 (ALOHA)   | 6     | ✓        | –        | 2023 |
| <b>GigaBrain-0.5M</b> [51] | 500K     | Multiple    | 200+  | ✓        | ✓        | 2025 |

Bimanual evaluation protocols remain less standardized. Most papers define their own task suites (*e.g.*,  $\pi_0$  [2] evaluates on laundry folding, box assembly, and table busing), making cross-method comparison difficult. Task completion criteria also vary: some papers use binary success, others use partial completion scores, and temporal efficiency is rarely reported.

**Generalization metrics** assess whether the VLA transfers to novel settings:

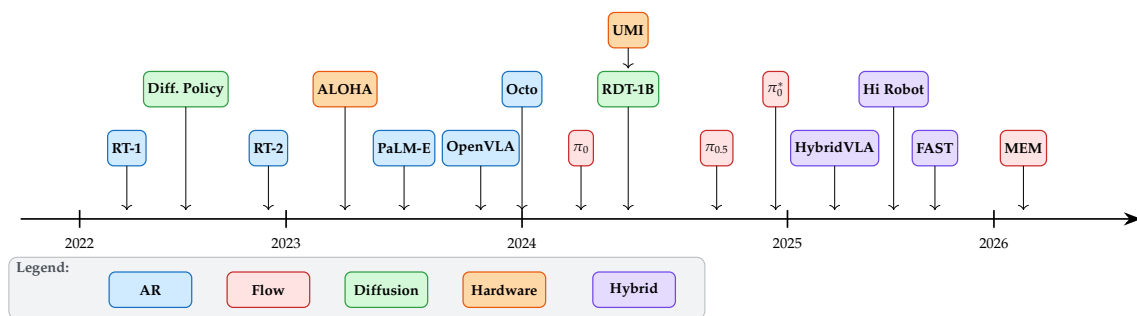
$$\text{Gen}(\pi_\theta) = \frac{\text{SR}_{\text{novel}}}{\text{SR}_{\text{train}}}, \quad (11)$$

where  $\text{SR}_{\text{novel}}$  and  $\text{SR}_{\text{train}}$  are success rates on novel and training environments respectively. A generalization ratio near 1.0 indicates robust transfer. **RT-2** [1] reported a generalization ratio of  $\sim 0.76$  for novel objects, while  $\pi_{0.5}$  [3] achieved  $\sim 0.95$  for novel homes, indicating strong environment generalization.

Bimanual-specific benchmarks remain limited: most simulation suites focus on single-arm tasks, and real-world bimanual evaluation varies across papers. This gap motivates the need for standardized bimanual benchmarks (Section 12).

## 5. VLA Architectures and Foundations

This section presents VLA architectures organized by their action generation mechanism. We identify four families: autoregressive, flow-based, diffusion-based, and hybrid. Figure 2 traces the chronological development of these methods from 2022 to 2025. Figure 1 provides an overview of the families and their constituent methods, while Figure 3 contrasts the four architectural paradigms side by side. Table 3 compares representative methods across key architectural dimensions.



**Figure 2.** Timeline of key VLA and bimanual manipulation milestones (2022–2026). Colors indicate the architectural family: autoregressive (blue), flow-based (red), diffusion-based (green), hardware platforms (orange), and hybrid/efficient methods (purple). The field has accelerated rapidly, with the majority of VLA contributions appearing in 2024–2026.

### 5.1. Autoregressive VLAs

Autoregressive VLAs generate actions by extending the VLM’s language modeling capability to action tokens. This approach directly builds on the pre-trained language model’s sequential generation ability.

### 5.1.1. RT-1 and RT-2

**RT-1** [60] was among the first Transformer-based robot policies trained on large-scale real-world data. It processes image histories through a FiLM-EfficientNet encoder and generates discretized actions via per-dimension classification heads. Trained on 130,000 demonstrations from Google’s mobile manipulator fleet, RT-1 achieved 97% success on seen tasks and 76% on unseen tasks, establishing that data diversity produces meaningful generalization. RT-1 does not address bimanual manipulation, but its lessons directly informed subsequent VLA designs.

**RT-2** [1] took the decisive step of unifying vision-language understanding and action generation within a single VLM. By fine-tuning a PaLI-X (55B parameters) or PaLM-E (12B parameters) model to output discretized actions as text tokens, RT-2 demonstrated that VLM pre-training on web data transfers to robotic manipulation. It exhibited capabilities absent from RT-1: reasoning about object categories, interpreting novel instructions, and performing rudimentary chain-of-thought planning for multi-step actions. Semantic knowledge encoded in VLM weights (object affordances, spatial relationships, physical intuition) directly benefits action generation. RT-2 remains closed-source, however, and its 55B-parameter scale demands TPU-class compute, putting real-time control out of reach for most research groups and limiting reproducibility.

### 5.1.2. OpenVLA

Reproducibility was a major barrier for VLA research until **OpenVLA** [5], the first fully open-source 7B-parameter VLA. Based on the Prismatic VLM architecture, it tokenizes actions into 256 discrete bins per dimension following RT-2’s approach and trains on the OXE dataset. Despite its open weights, OpenVLA matches **RT-2-X** [61] (the cross-embodiment variant of RT-2 trained on OXE alongside **RT-1-X** [62]). Its release catalyzed community research and revealed important scaling behaviors: performance improves consistently with data diversity, and fine-tuning on small task-specific datasets yields substantial gains over the pre-trained checkpoint. A subsequent update, **OpenVLA 2.0** [63], improved data curation and action tokenization, narrowing the gap to proprietary systems.

### 5.1.3. Octo

Although Octo uses a diffusion action head, we discuss it here because it was designed as a generalist initialization for downstream fine-tuning, a role shared with autoregressive VLAs. Where OpenVLA adapts an existing VLM, **Octo** [6] takes a different path: a purpose-built 93M-parameter Transformer with a diffusion action head, bridging the autoregressive and diffusion approaches. Prompted with language instructions or goal images, Octo supports flexible action spaces and was trained on 800,000 episodes from OXE. It serves as a versatile initialization for fine-tuning on downstream tasks, including bimanual manipulation with ALOHA. Because it uses a diffusion action head rather than autoregressive token prediction, Octo straddles both categories in our taxonomy (Figure 1). Its modest parameter count, however, restricts capacity for complex reasoning compared to VLM-based approaches.

Other autoregressive VLAs include **GR-1** [64] (video prediction as an implicit world model), **HAMSTER** [65] (unified vision-language-action prediction), **SpatialVLA** [66] (explicit spatial representations), **BAKU** [67] (efficient multi-task architecture), **KAT** [68] (keypoint-action tokens), and **SimpleVLA** [69] (minimal design matching elaborate systems with RL fine-tuning).

## 5.2. Flow-Based VLAs

Flow-based VLAs use flow matching (Section 2.3) to generate continuous action chunks, avoiding the discretization bottleneck of autoregressive approaches. As shown in Figure 3(b), the flow head iteratively denoises a noise sample conditioned on VLM features.

### 5.2.1. $\pi_0$

The discretization bottleneck of autoregressive VLAs motivated a structurally different action head.  $\pi_0$  [2] addresses it with the first flow-matching action head for VLAs. Built on a 3B-parameter PaLI Gemma [25] backbone, the model processes image and language tokens through the VLM, then uses the resulting hidden states to condition a flow-matching network that generates action chunks. The action head consists of Transformer layers that jointly attend to VLM features and noisy action tokens, iteratively denoising over  $K = 10$  flow steps.

$\pi_0$  achieved state-of-the-art results on bimanual dexterous tasks including laundry folding (80% success), box assembly, and table busing. The flow-matching formulation is critical for bimanual tasks: it generates smooth, coherent 16-dimensional action chunks (8 per arm) without the quantization artifacts of autoregressive methods or the slow sampling of diffusion models. Pre-training on OXE followed by task-specific fine-tuning proved essential; the pre-trained model provides a strong initialization that allows learning from relatively few bimanual demonstrations. A significant limitation is that  $\pi_0$ 's strongest results depend on proprietary multi-task data collected across Physical Intelligence's robot fleet; reproducing these results with publicly available data alone has not been demonstrated, raising questions about how much of the performance stems from architecture versus data advantage.

### 5.2.2. $\pi_{0.5}$

$\pi_{0.5}$  [3] extends  $\pi_0$  to open-world deployment by introducing a hierarchical architecture. A high-level VLM generates subgoal language commands, while the low-level flow-matching policy executes motor actions. This decomposition allows  $\pi_{0.5}$  to handle complex, multi-step household tasks such as "clean the kitchen" that require planning over minutes rather than seconds.

$\pi_{0.5}$  was deployed on a fleet of mobile manipulators in real homes, where it reached a 98% success rate at following verbal instructions under a controlled set of household tasks, a notable demonstration of VLA generalization beyond the lab, though the result is self-reported under conditions chosen by the authors and has not been independently reproduced. The hierarchical architecture proves well suited to bimanual tasks, where the high-level model can decompose complex instructions into single-step bimanual primitives.

### 5.2.3. $\pi_0^*$ and RECAP

$\pi_0^*$  (also referred to as  $\pi_{0.6}^*$  in the original publication) [4] addresses a fundamental limitation of imitation learning: performance is bounded by the quality of demonstrations.  $\pi_0^*$  introduces RECAP (Reinforcement Learning from Autonomous CAPability), a training approach in which the VLA collects experience autonomously and then improves via reinforcement learning. A VLM-based evaluator provides success/failure labels, eliminating hand-designed reward functions. Starting from a  $\pi_0$  checkpoint, RECAP alternates between autonomous data collection and policy optimization, progressively improving beyond the demonstration distribution. On bimanual tasks,  $\pi_0^*$  achieved 10–40% absolute improvement over the demonstration-only baseline, demonstrating that VLAs can bootstrap their own improvement.

## 5.3. Diffusion-Based VLAs

Diffusion-based approaches apply denoising diffusion probabilistic models to action generation. The key tradeoff is generation quality versus inference latency.

### 5.3.1. Diffusion Policy

The idea of treating action generation as a denoising process originated with **Diffusion Policy** [40], which established that generative models outperform deterministic behavioral cloning for tasks with multimodal demonstrations. The principal drawback is computational cost:  $K = 50$ – $100$  DDPM steps push latency to  $\sim 300$  ms per chunk, making real-time bimanual control impractical without acceleration (DDIM, consistency distillation). While Diffusion Policy predates VLAs (it uses task-specific encoders),

its innovations (denoising action chunks, classifier-free guidance, temporal ensembling) are reused across subsequent VLA architectures.

### 5.3.2. RDT-1B

A natural question is whether scaling alone can close the gap between diffusion and flow-based VLAs. **RDT-1B** [44] tests this hypothesis by pushing diffusion-based action generation to 1.2 billion parameters, creating a “diffusion foundation model” for bimanual manipulation. The model uses a Transformer backbone (inspired by DiT) to denoise action chunks, conditioned on visual features from a pre-trained vision encoder and language features from a pre-trained text encoder. Pre-trained on multi-robot datasets and fine-tuned on ALOHA bimanual tasks, RDT-1B confirms that scale benefits diffusion-based robot policies just as it benefits language models.

RDT-1B’s large parameter count allows it to capture the complex coordination patterns required for bimanual tasks. On ALOHA benchmarks, it outperformed ACT and Diffusion Policy on tasks requiring tight bimanual coordination such as handovers and collaborative assembly. The model also exhibited improved robustness to visual distractors and perturbations compared to smaller diffusion policies.

RDT-1B uses separate vision and language encoders (SigLIP and T5) rather than a unified VLM, reaching competitive performance through scale rather than joint pre-training. The DiT backbone handles long bimanual sequences ( $64 \text{ timesteps} \times 16 \text{ dimensions} = 1024 \text{ elements}$ ) efficiently.

### 5.3.3. CogACT

A key tension in diffusion VLAs is the conflict between language generation and action denoising losses. **CogACT** [70] resolves this by introducing learned “cognitive action tokens” that bridge VLM semantic representations and the diffusion action head. This abstraction layer isolates the two objectives, reducing their interference during training.

Related approaches include **PerAct** [71] (3D voxel-based manipulation), **RVT** [72] (efficient multi-view 3D manipulation), **3D Diffusion Policy** [73] (diffusion over point clouds), and **Transfusion** [74] (unified next-token prediction with diffusion generation).

## 5.4. Hybrid and Efficient VLAs

Several recent architectures combine multiple action generation approaches or focus on computational efficiency.

### 5.4.1. HybridVLA

Autoregressive and flow-based approaches have complementary strengths: autoregressive models excel at discrete decisions (*e.g.*, grasp vs. release) while flow models excel at continuous trajectories. **HybridVLA** [75] exploits this complementarity by integrating both within a unified architecture, routing discrete action components through an autoregressive head and continuous components through a flow-matching head. This hybrid design outperforms either approach alone on tasks with mixed discrete-continuous action spaces.

The hybrid architecture suits bimanual manipulation well: gripper commands (open/close) are inherently discrete while arm motions are continuous. HybridVLA routes each component to the appropriate head, with both conditioning on the same VLM hidden states. A limitation is added training complexity: balancing the two heads requires careful loss weighting, and no published ablation isolates each head’s independent contribution.

### 5.4.2. TinyVLA

Inference latency remains the primary obstacle to deploying large VLAs on real-time bimanual systems. **TinyVLA** [7] tackles this through knowledge distillation: a compact student model is trained to match a full-size VLA’s outputs, preserving task performance while reducing inference time. The result is real-time control at 50 Hz on consumer GPUs, making VLA deployment practical for

bimanual systems with tight latency requirements. The distillation step introduces a performance ceiling, however: the student cannot exceed its teacher, and the fidelity of distillation degrades for high-dimensional bimanual action distributions.

#### 5.4.3. MiniVLA

A contrasting strategy to distillation is to design for efficiency from the outset. **MiniVLA** [76] pairs a small VLM backbone with a lightweight action head, attaining competitive performance on standard benchmarks at a fraction of the compute. This ground-up efficiency could benefit bimanual systems, though MiniVLA has not been evaluated on such tasks. MiniVLA’s reduced capacity limits its ability to handle complex multi-step reasoning.

#### 5.4.4. FAST

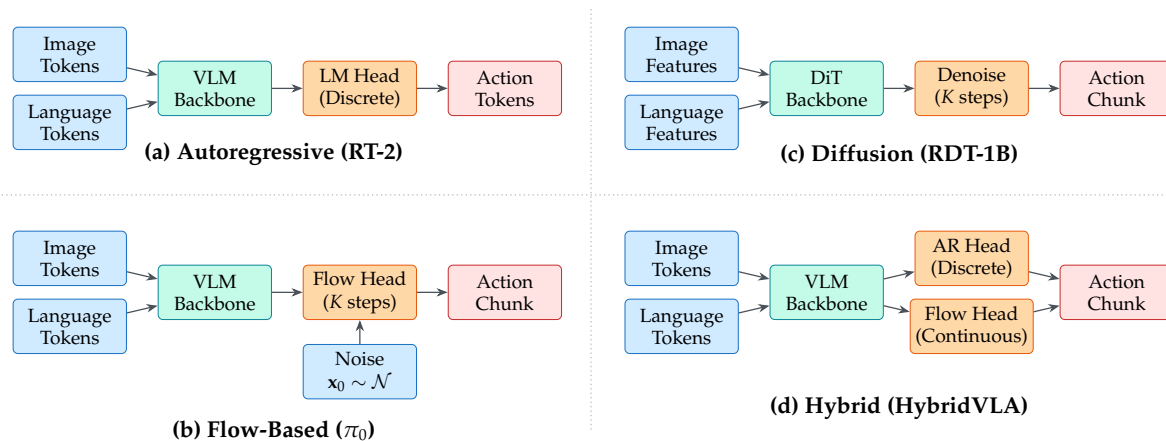
Efficient autoregressive action generation requires a better tokenization scheme than naive per-dimension binning. **FAST** [8] provides one with a learned action tokenizer that compresses continuous action chunks into discrete tokens. A VQ-VAE first learns to tokenize action chunks, and then a VLM is fine-tuned to predict these tokens. The tokenizer captures robot-specific action structure (temporal correlations, joint coupling), producing a compact discrete representation that avoids the quantization artifacts of naive binning. FAST achieves state-of-the-art results among autoregressive VLAs while maintaining the simplicity of text-token generation.

Applied to dual-arm control, the FAST tokenizer learns coordinated patterns as single codebook entries, compressing a 50-step chunk ( $50 \times 16 = 800$  values) into 16–32 tokens (25–50 $\times$  compression), which reduces latency and improves long-range dependency modeling. A limitation is that the VQ-VAE codebook is fixed after training, so novel action patterns outside the codebook’s coverage may be poorly represented.

**Hi Robot** [29] takes the hierarchical route: a high-level VLM selects subtasks while a low-level flow policy executes them, enabling open-ended instruction following on bimanual platforms.

**Table 3.** Comparison of VLA architectures. Action type indicates the generation mechanism: AR = autoregressive, FM = flow matching, Diff = diffusion, Hybrid = combined. Bimanual indicates demonstrated bimanual capability. Params refers to the total model size.

| Method                | Action Type       | VLM Backbone  | Params | Chunk $H$ | Bimanual | Open-Source | Year |
|-----------------------|-------------------|---------------|--------|-----------|----------|-------------|------|
| <b>RT-1</b> [60]      | AR (discrete)     | EfficientNet  | 35M    | 1         | –        | –           | 2022 |
| <b>RT-2</b> [1]       | AR (discrete)     | PaLI-X/PaLM-E | 55B    | 1         | –        | –           | 2023 |
| <b>Octo</b> [6]       | Diff head         | Custom        | 93M    | 4         | ✓        | ✓           | 2024 |
| <b>OpenVLA</b> [5]    | AR (discrete)     | Prismatic     | 7B     | 1         | –        | ✓           | 2024 |
| $\pi_0$ [2]           | FM                | PaLIGemma     | 3B     | 50        | ✓        | –           | 2024 |
| $\pi_{0.5}$ [3]       | FM (hierarchical) | PaLIGemma     | 3B     | 50        | ✓        | –           | 2025 |
| $\pi_0^*$ [4]         | FM + RL           | PaLIGemma     | 3B     | 50        | ✓        | –           | 2025 |
| <b>RDT-1B</b> [44]    | Diff (DiT)        | SigLIP + T5   | 1.2B   | 64        | ✓        | ✓           | 2024 |
| <b>CogACT</b> [70]    | Diff head         | VLM           | 7B     | 16        | –        | ✓           | 2025 |
| <b>HybridVLA</b> [75] | AR + FM           | VLM           | 7B     | 50        | ✓        | –           | 2025 |
| <b>TinyVLA</b> [7]    | Distilled         | Small VLM     | 1B     | 10        | –        | ✓           | 2024 |
| <b>MiniVLA</b> [76]   | Efficient head    | Small VLM     | 300M   | 8         | –        | ✓           | 2024 |
| <b>FAST</b> [8]       | AR (learned tok.) | VLM           | 7B     | 50        | –        | ✓           | 2025 |
| <b>Hi Robot</b> [29]  | Hierarchical FM   | VLM           | 3B+    | 50        | ✓        | –           | 2025 |



**Figure 3.** Architectural comparison of the four VLA families. (a) Autoregressive VLAs (RT-2, OpenVLA) discretize actions and generate them as language tokens. (b) Flow-based VLAs ( $\pi_0$ ) use a flow-matching head that iteratively denoises a noise sample conditioned on VLM features. (c) Diffusion VLAs (RDT-1B) use a Diffusion Transformer to denoise action chunks. (d) Hybrid VLAs (HybridVLA) combine autoregressive and flow heads for discrete and continuous action components, respectively.

Table 4 compares VLA methods across standard benchmarks. Three findings stand out from this comparison. First, flow-based VLAs ( $\pi_0$ ) dominate across all benchmarks, with notably large margins on long-horizon tasks (LIBERO-Long) where action chunk coherence matters most. Second, hybrid approaches (**HybridVLA**) rank second, suggesting that combining autoregressive and flow-based generation captures complementary strengths. Third, efficient models (**TinyVLA**) trade 10–15% performance for significantly reduced compute, a worthwhile tradeoff for resource-constrained deployments.

**Table 4.** VLA performance on standard benchmarks. LIBERO scores are normalized success rates averaged across task suites. Bridge V2 reports average success rate across evaluation tasks. SIMPLER reports success rate on the Google Robot evaluation suite. Bold indicates best in column.<sup>1</sup>

| Method         | LIBERO-Spatial | LIBERO-Object | LIBERO-Goal | LIBERO-Long | Bridge V2   | SIMPLER     |
|----------------|----------------|---------------|-------------|-------------|-------------|-------------|
| RT-1 [60]      | –              | –             | –           | –           | 45.2        | –           |
| RT-2 [1]       | –              | –             | –           | –           | 52.1        | 55.3        |
| Octo [6]       | 78.9           | 85.7          | 72.1        | 46.3        | 54.8        | 48.7        |
| OpenVLA [5]    | 84.7           | 88.4          | 79.2        | 53.7        | 58.3        | 56.2        |
| $\pi_0$ [2]    | <b>96.2</b>    | <b>97.1</b>   | <b>93.5</b> | <b>78.4</b> | <b>72.6</b> | <b>71.3</b> |
| RDT-1B [44]    | 89.3           | 91.5          | 84.7        | 62.1        | 61.4        | –           |
| CogACT [70]    | 87.1           | 89.8          | 81.3        | 58.9        | 59.7        | –           |
| FAST [8]       | 91.5           | 93.2          | 86.8        | 65.3        | 64.2        | 62.8        |
| HybridVLA [75] | 93.8           | 95.4          | 90.1        | 72.6        | 68.9        | 67.4        |
| TinyVLA [7]    | 82.3           | 86.1          | 75.4        | 51.2        | 55.6        | 52.1        |

Autoregressive VLAs benefit most from VLM pre-training but struggle with high-dimensional continuous actions. Flow-based VLAs offer the best quality-speed balance for bimanual tasks. Diffusion-based VLAs provide strong multimodal modeling at higher cost. Architecture alone does not determine performance; training strategy is equally decisive. Two modular extensions apply across all families: *memory modules* [77–79] for long-horizon task tracking (Section 8.4) and *world models* [80–82] for future state prediction (Section 11.4).

## 6. Training Recipes and Data Strategies

The performance of a VLA depends as much on how it is trained as on its architecture. This section reviews the three-stage training pipeline (pre-training, post-training, and reinforcement learning) as well as data collection strategies for bimanual manipulation. Figure 9 illustrates the complete pipeline from VLM pre-training through deployment.

### 6.1. Pre-training

As outlined in the “Training” branch of Figure 1, VLA pre-training proceeds in two phases. First, the VLM backbone is pre-trained on internet-scale image-text data, learning general visual and semantic representations. Second, the full VLA (backbone plus action head) is trained on large-scale robot demonstration data.

The VLM pre-training phase builds on existing VLM checkpoints.  $\pi_0$  [2] initializes from PaLI Gemma [25], a 3B-parameter VLM pre-trained on web-scale image-text data using the SigLIP visual encoder and Gemma [26] language backbone. **OpenVLA** [5] initializes from Prismatic, a 7B-parameter VLM combining DINOv2 and SigLIP visual encoders with a Llama-2 language backbone. **RT-2** [1] initializes from PaLI-X, a 55B-parameter VLM. This initialization provides the VLA with broad visual understanding, language comprehension, and spatial reasoning capabilities before it encounters any robot data.

The trend is toward smaller, efficient VLM backbones paired with powerful action heads: larger VLMs encode richer representations but are too slow for real-time control, while 3B-parameter models like PaLI Gemma allow flow-matching heads with multiple forward passes per chunk.

The robot data pre-training phase trains the VLA on diverse cross-embodiment data.  $\pi_0$  [2] pre-trains on a mixture of OXE data and proprietary multi-task data spanning seven robot embodiments and hundreds of tasks, using a co-training objective that mixes action prediction with VLM text generation to preserve language capabilities. The co-training loss is:

$$\mathcal{L}_{\text{co-train}} = \lambda_{\text{act}} \mathcal{L}_{\text{FM}} + \lambda_{\text{lang}} \mathcal{L}_{\text{LM}}, \quad (12)$$

where  $\mathcal{L}_{\text{FM}}$  is the flow matching loss (Equation 5),  $\mathcal{L}_{\text{LM}}$  is the language modeling loss, and  $\lambda_{\text{act}}, \lambda_{\text{lang}}$  are balancing weights. **OpenVLA** [5] pre-trains exclusively on OXE data, demonstrating that publicly available data suffices for competitive pre-training. The key finding across all approaches is that data diversity (spanning multiple embodiments, environments, and tasks) matters more than dataset size for downstream generalization. This diversity also benefits memory-augmented VLAs [77,78]: pre-training on varied multi-step tasks exposes the memory module to diverse temporal patterns, improving its ability to track long-horizon bimanual task state. Additionally, world models can serve as *data engines* during pre-training; **GigaWorld-0** [83] generates synthetic robot episodes via video generation and sim-to-real transfer, augmenting real demonstration data at scale. However, no published study has rigorously ablated whether pre-training gains stem from data diversity or backbone quality, making it difficult to attribute improvements to either factor alone.

### 6.2. Post-training and Fine-tuning

Post-training adapts a pre-trained VLA to specific tasks, embodiments, or environments. Fine-tuning on task-specific demonstrations is the most common approach.  $\pi_0$  [2] fine-tunes on 50–200 bimanual demonstrations per task, which yields strong performance from a well-initialized model. **OpenVLA** [5] demonstrated that fine-tuning on as few as 10 demonstrations can yield significant improvements on in-distribution tasks.

Co-training (mixing target-task data with pre-training data during fine-tuning) prevents catastrophic forgetting and often improves performance on the target task.  $\pi_0$  [2] mixes cross-embodiment data with single-task data throughout fine-tuning, finding that this consistently outperforms fine-tuning on target data alone. The intuition is that cross-embodiment data provides a regularization effect, preventing the model from overfitting to the small fine-tuning dataset.

The mixing ratio is a key hyperparameter:  $\pi_0$  [2] uses a 1:1 ratio, while **Mobile ALOHA** [42] found even 10% diverse co-training data improves bimanual success. Tasks dissimilar to pre-training benefit from more target data; visually similar tasks benefit from more diverse co-training.

Parameter-efficient fine-tuning via Low-Rank Adaptation (LoRA) [84] offers an alternative to full fine-tuning: by injecting low-rank weight updates into frozen VLM layers, LoRA reduces GPU memory requirements while preserving pre-trained representations. Preference-based optimization

methods such as DPO [85] have also been explored for aligning VLA outputs with human preferences, though their application to bimanual manipulation remains limited.

**Knowledge Insulation** [86] freezes certain VLM layers during fine-tuning to preserve language understanding. **Align-then-Steer** [87] uses two-phase alignment and constrained fine-tuning. **RoboMimic** [88] finds that observation representations and data quality have outsized impact on offline IL, and **RoboAgent** [89] improves generalization through semantic augmentations.

### 6.3. Reinforcement Learning for VLAs

Imitation learning alone caps VLA performance at the demonstration quality. Reinforcement learning (RL) offers a path to surpass this ceiling. Early large-scale RL for robotic manipulation (**QT-Opt** [90] trained grasping policies from over 500,000 real grasps) showed that RL can scale in the real world. Offline RL methods such as **Conservative Q-Learning (CQL)** [91] and **Advantage-Weighted Regression (AWR)** [92] learn from static datasets without further environment interaction, providing a lower-risk alternative (see Levine *et al.* [93] for a tutorial). Policy gradient methods such as PPO [94] remain the most common choice for online fine-tuning of VLAs.

Online RL for VLAs was first realized by  $\pi_0^*$  (**RECAP**) [4]. The RECAP pipeline works as follows: (1) the VLA policy autonomously executes tasks in the real world, (2) a VLM-based evaluator labels each episode as success or failure, (3) successful episodes are added to the training set and failures are discarded, and (4) the VLA is fine-tuned on the augmented dataset. This cycle repeats, gradually expanding the VLA's competence beyond the original demonstration distribution.

RECAP's key innovation is using a separate VLM as an autonomous reward function, eliminating the need for hand-designed reward signals. On bimanual tasks, RECAP improved laundry folding success from 60% (demonstration-only) to over 90% after several cycles of autonomous practice. This demonstrates that VLAs can self-improve on bimanual tasks, a necessary capability for scaling deployment. Algorithm 1 summarizes the RECAP pipeline.

---

#### Algorithm 1 RECAP: RL from Autonomous Capability

---

**Require:** Pre-trained VLA policy  $\pi_\theta$ , VLM evaluator  $\mathcal{V}$ , task set  $\mathcal{T}$

**Require:** Demonstration buffer  $\mathcal{D}_{\text{demo}}$ , practice buffer  $\mathcal{D}_{\text{auto}} \leftarrow \emptyset$

```

1: for cycle  $c = 1, 2, \dots, C$  do
2:   // Autonomous data collection
3:   for task  $\tau \in \mathcal{T}$  do
4:     Execute  $\pi_\theta$  on  $\tau$ , collect trajectory  $\xi$ 
5:     Evaluate:  $r \leftarrow \mathcal{V}(\xi, \tau)$  // VLM judges success
6:     if  $r = \text{success}$  then
7:        $\mathcal{D}_{\text{auto}} \leftarrow \mathcal{D}_{\text{auto}} \cup \{\xi\}$ 
8:     end if
9:   end for
10:  // Policy improvement
11:  Fine-tune  $\pi_\theta$  on  $\mathcal{D}_{\text{demo}} \cup \mathcal{D}_{\text{auto}}$ 
12: end for
13: return Improved policy  $\pi_\theta$ 

```

---

A different RL objective is speed rather than success: human teleoperation of bimanual systems is typically slow and cautious, so policies that merely imitate inherit this inefficiency. **SAIL** [95] addresses this by training VLA policies to execute tasks *faster* than the demonstrations. Using time-warped demonstrations and a reward that encourages speed while maintaining success, SAIL produces policies that complete bimanual tasks in less time than human demonstrators, removing the teleoperation speed bottleneck.

Other RL approaches include **VLA-RL** [96] (policy gradients on the action head), **Con-RFT** [97] (consistency-regularized RL), **Q-Transformer** [98] (offline RL via autoregressive Q-functions), **DPPO** [99] (policy optimization for diffusion policies), and **Self-Improving Foundation Models** [100] (autonomous data generation and filtering).

Rather than learning purely from collected experience, **GigaBrain-0.5M** [51] integrates a world model with VLA training. The world model predicts future visual observations given current actions, allowing the VLA to “imagine” the consequences of action sequences without physical execution, a form of model-based RL that generates synthetic training data. Two-arm coordination demands that the world model predict complex multi-body dynamics, including how both arms and the manipulated object evolve over time. While world-model-based RL for bimanual VLAs is still early-stage, it represents a promising path toward sample-efficient training of complex coordination behaviors.

#### 6.4. Data Collection for Bimanual Manipulation

High-quality bimanual demonstration data is the bottleneck for VLA training. Three data collection strategies have emerged.

**ALOHA** [17] uses bilateral teleoperation: a human operator controls follower arms by physically moving kinematically identical leader arms. This provides intuitive, low-latency control for dexterous bimanual tasks. The ALOHA hardware costs under USD 20,000 and has been replicated at dozens of institutions, creating a growing ecosystem of bimanual data.

**UMI** [43] decouples data collection from the robot entirely. Operators use hand-held gripper tools with visual-inertial tracking to demonstrate tasks in any environment. The collected trajectories are retargeted to the robot’s action space during training. UMI allows data collection by non-experts in diverse settings, increasing data diversity.

**Autonomous data collection**, as in RECAP [4], uses the VLA itself to collect additional data. Starting from a reasonably capable policy, the robot attempts tasks autonomously, and a success classifier (often another VLM) labels the outcomes. This approach can generate thousands of additional episodes with minimal human effort, and the data naturally covers the policy’s distribution, reducing the train-test mismatch that plagues behavioral cloning.

**Data quality** matters as much as quantity: teleoperation quality varies across operators, language labels are often inconsistent, and demonstrations including recovery from near-failures contribute disproportionately to robustness. Fleet-based data strategies [101] established the precedent for VLA-scale collection, and **GigaBrain-0.5M** [51] addresses quality through automated world-model-based filtering.

Data collection throughput varies: ALOHA teleoperation [17] yields 50–100 demos/hour, UMI [43] achieves ~110 demos/hour with non-expert operators (3× faster than spacemouse), and autonomous RECAP [4] runs continuously without supervision (4–12 episodes/hour for complex tasks). These economics favor a hybrid strategy: bootstrap with demonstrations, then scale through autonomous practice.

#### 6.5. Data Scaling Laws

VLA performance scales differently with different data types [102]. **Cross-embodiment pre-training data** exhibits log-linear scaling, with dataset diversity (number of embodiments, environments, tasks) at least as important as raw volume [5]. **Task-specific fine-tuning data** shows steep initial scaling and diminishing returns: simple tasks require as little as 5 hours, while complex bimanual tasks need 100+ hours [2]. **Autonomous practice data** from RECAP [4] yields large gains from modest data: ~300 autonomous trajectories per iteration improved laundry folding from ~30% to over 90%.

Table 5 summarizes the training strategies across representative VLA methods. The pipeline is converging on a three-stage recipe: (1) initialize from a strong VLM, (2) pre-train on diverse cross-embodiment data with co-training, and (3) fine-tune on task-specific bimanual demonstrations, optionally followed by RL. How actions are represented and executed within these pipelines is equally consequential for real-time bimanual control.

**Table 5.** Comparison of training strategies for VLA models. Pre-train data and fine-tune data indicate the primary datasets used. RL indicates whether reinforcement learning is incorporated.

| Method          | VLM Init     | Pre-train Data    | Fine-tune Data     | Co-train | RL | Key Strategy          |
|-----------------|--------------|-------------------|--------------------|----------|----|-----------------------|
| RT-2 [1]        | PaLI-X       | Google fleet      | –                  | –        | –  | VLM co-fine-tuning    |
| OpenVLA [5]     | Prismatic    | OXE               | Task-specific      | –        | –  | Open data pre-train   |
| Octo [6]        | From scratch | OXE (800K)        | Task-specific      | –        | –  | Cross-embodiment init |
| $\pi_0$ [2]     | PaLIGemma    | OXE + proprietary | 50–200 demos       | ✓        | –  | Co-training mix       |
| $\pi_{0.5}$ [3] | PaLIGemma    | OXE + fleet       | Fleet demos        | ✓        | –  | Hierarchical training |
| $\pi_0^*$ [4]   | $\pi_0$ ckpt | Same as $\pi_0$   | Autonomous + demos | ✓        | ✓  | RECAP (VLM reward)    |
| RDT-1B [44]     | SigLIP+T5    | Multi-robot       | ALOHA tasks        | –        | –  | Scale (1.2B params)   |
| FAST [8]        | VLM          | OXE               | Task-specific      | –        | –  | Learned tokenizer     |
| GigaBrain [51]  | VLM          | GigaBrain-0.5M    | Task-specific      | –        | ✓  | World-model RL        |

## 7. Action Representations and Real-Time Execution

The choice of action representation directly shapes VLA performance, especially for bimanual tasks where the action space is high-dimensional and temporal coordination is critical. Building on the architectural foundations in Section 5 and the “Actions” branch of the taxonomy in Figure 1, this section covers discrete tokenization, continuous generation, action chunking strategies, and recent advances in real-time execution.

### 7.1. Discrete Action Tokenization

The simplest approach to interfacing robot actions with a language model is to discretize continuous actions into tokens.

**Uniform binning**, used by RT-2 [1] and OpenVLA [5], divides each action dimension into  $B$  equally spaced bins (typically  $B = 256$ ). A 7-DOF arm action is represented as 7 tokens, each from a vocabulary of size  $B$ . This approach is simple but introduces quantization error proportional to  $1/B$  and requires sequential token generation, with latency scaling linearly with the number of dimensions. For bimanual systems with  $d_a = 16$  (see Section 2.4), generating 16 tokens sequentially becomes a latency bottleneck.

The inefficiency of uniform binning motivated a learned alternative. In FAST [8], a VQ-VAE is trained to compress action chunks into a small number of discrete tokens. Because the tokenizer learns the structure of robot actions (temporal smoothness, joint correlations, coordination patterns), it can represent a 50-step, 16-dimensional bimanual action chunk with as few as 32 tokens, compared to  $50 \times 16 = 800$  tokens with per-dimension binning. The learned tokenizer also produces a denser token vocabulary where every token corresponds to a meaningful action pattern, unlike uniform binning where most of the 256 bins are rarely used.

Cross-embodiment transfer demands that action tokens generalize across morphologies. **Universal action tokenization** [103] achieves this with a shared tokenizer that learns embodiment-agnostic action representations, so that a single autoregressive VLA can generate actions for robots with different morphologies. For bimanual systems, universal tokenization offers the prospect of transferring manipulation knowledge from single-arm datasets to dual-arm systems, since the tokenizer can learn correspondences between single-arm and bimanual action patterns.

Complementary approaches include **consistency models** [104] (distilling multi-step diffusion into a single forward pass), **ACT** [105] (improved temporal ensembling for bimanual trajectories), and **RACER** [106] (language-guided corrective actions for error recovery).

Discrete tokenization preserves compatibility with VLM text generation but introduces quantization error that compounds across 16 bimanual action dimensions and 50 timesteps per chunk. Continuous generation avoids this issue but requires iterative denoising.

### 7.2. Continuous Action Generation

Continuous action generation avoids discretization entirely, predicting real-valued action vectors.

Flow matching ( $\pi_0$  [2]) and diffusion (**Diffusion Policy** [40], **RDT-1B** [44]), whose architectural details appear in Section 5, generate continuous actions through iterative denoising. The primary

advantage is fidelity: continuous predictions avoid quantization error, which compounds across the 16 action dimensions of a bimanual system. The primary disadvantage is inference cost: each prediction requires  $K$  denoising steps, each involving a forward pass through the action head.

The number of denoising steps  $K$  trades off quality against speed.  $\pi_0$  [2] uses  $K = 10$  flow matching steps, which strikes a good balance for bimanual control at 50 Hz. Diffusion policies typically require  $K = 50$ – $100$  DDPM steps, though DDIM acceleration reduces this to  $K = 10$ – $20$  with modest quality loss.

### 7.3. Action Chunking Strategies

Action chunking (Section 2.2) is now standard in VLA systems. The chunk horizon  $H$  is a critical hyperparameter.

Short chunks ( $H = 1$ – $4$ ) provide high reactivity (the policy can respond to environmental changes quickly) but require frequent VLM inference and suffer from myopic behavior. **Octo** [6] uses  $H = 4$ , suitable for its lightweight architecture.

Long chunks ( $H = 50$ – $100$ ) amortize inference cost and capture long-range temporal structure, but commit the robot to extended open-loop execution.  $\pi_0$  [2] uses  $H = 50$ , which at 50 Hz corresponds to 1 second of motion. When both arms act in concert (*e.g.*, folding), long chunks capture the coordinated motion pattern of both arms within a single prediction.

When a task has distinct phases (approach, grasp, manipulate, release), the chunk horizon should cover at least one complete phase. Empirically,  $H = 50$  at 50 Hz (1 second) is the sweet spot for most bimanual primitives.

**Temporal ensembling** blends overlapping action chunks to smooth transitions and improve robustness. Given the current chunk  $\mathbf{A}_t$  and the previous chunk  $\mathbf{A}_{t-s}$  (shifted by  $s$  steps), the executed action is:

$$\hat{\mathbf{a}}_t = \lambda \mathbf{a}_t^{\text{new}} + (1 - \lambda) \mathbf{a}_t^{\text{old}}, \quad (13)$$

where  $\mathbf{a}_t^{\text{new}}$  is the action at time  $t$  from the most recently predicted chunk  $\mathbf{A}_t$ ,  $\mathbf{a}_t^{\text{old}}$  is the corresponding action from the previously predicted (overlapping) chunk  $\mathbf{A}_{t-s}$ , and  $\lambda \in [0, 1]$  controls the blending weight, typically decaying exponentially over the chunk.

### 7.4. Real-Time Chunking (RTC)

Long action chunks and fast reactions seem fundamentally at odds, but **RTC** [107] resolves this tension by restructuring how chunks are generated and executed. Rather than computing an entire chunk and executing it open-loop, RTC interleaves computation and execution: while the current chunk is being executed, the next chunk is computed in the background. When the next chunk is ready, the policy smoothly transitions to it, regardless of where execution is in the current chunk.

RTC's key contribution is showing that with careful scheduling, a flow-matching VLA can achieve both long-horizon coherence (from large chunks) and sub-100 ms reactivity (from overlapping computation). In dual-arm tasks, this allows the robot to respond to unexpected perturbations (*e.g.*, an object slipping from one gripper) without waiting for the current chunk to complete.

The effective reaction time is bounded by the computation time for one chunk (typically 50–70 ms for  $\pi_0$ ), a significant improvement over the full chunk execution time (1000 ms for  $H = 50$  at 50 Hz).

### 7.5. Bidirectional Decoding (BID)

For tasks where both endpoints are well-defined but the intermediate trajectory is ambiguous, **BID** [108] generates action chunks from both ends simultaneously. Given a chunk of horizon  $H$ , two decoders are initialized: a forward decoder starting from the current state and a backward decoder starting from the goal state. The two decoders produce action sequences that are merged at a midpoint. This bidirectional approach is especially relevant for bimanual handover tasks, where the initial grasp (forward) and the final placement (backward) are well-defined but the intermediate transfer motion admits multiple solutions. BID resolves this ambiguity by anchoring both endpoints.

### 7.6. Training-Time Action Conditioning

A subtle source of performance degradation is the train-test mismatch in action conditioning: during training, the action head conditions on ground-truth features unavailable at test time. **TTAC** [109] bridges this gap by conditioning the action head on its own predictions during training via a stop-gradient mechanism. On bimanual tasks, this improves success rates by 5–15% without architectural changes, with gains strongest at high action dimensionality ( $d_a = 16$ ).

Table 6 summarizes the action representation landscape. Bimanual manipulation exposes the practical consequences of these architectural and training choices.

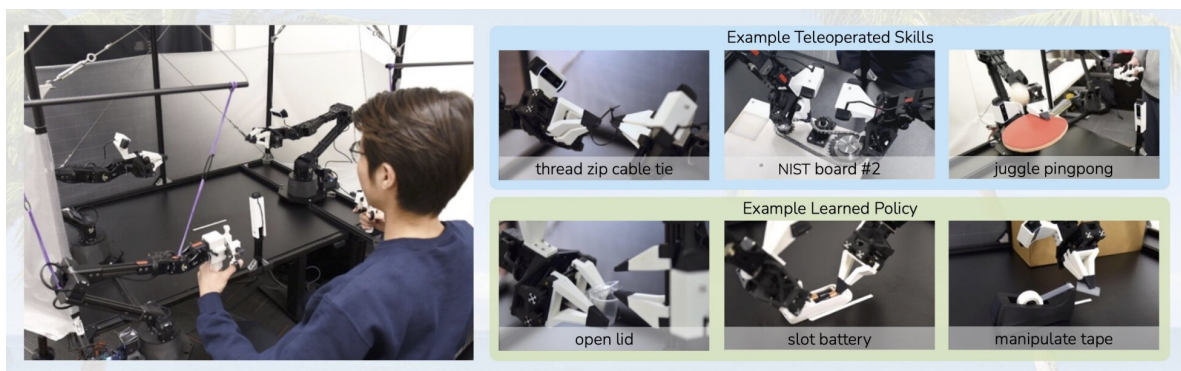
**Table 6.** Comparison of action representations in VLA models. Latency is per action-chunk inference on a single GPU. Bimanual  $d_a$  indicates the action dimension for bimanual systems.

| Method                   | Representation     | Chunk $H$ | Steps $K$ | Bimanual $d_a$ | Latency | Key Innovation       |
|--------------------------|--------------------|-----------|-----------|----------------|---------|----------------------|
| <b>RT-2</b> [1]          | Uniform bins (256) | 1         | –         | 8              | ~200 ms | VLM token reuse      |
| <b>OpenVLA</b> [5]       | Uniform bins (256) | 1         | –         | 8              | ~150 ms | Open-source          |
| <b>FAST</b> [8]          | Learned VQ-VAE     | 50        | –         | 16             | ~80 ms  | Compressed tokens    |
| $\pi_0$ [2]              | Flow matching      | 50        | 10        | 16             | ~70 ms  | VLM-conditioned flow |
| <b>Diff. Policy</b> [40] | DDPM/DDIM          | 16        | 50–100    | 16             | ~300 ms | Multimodal actions   |
| <b>RDT-1B</b> [44]       | DiT diffusion      | 64        | 20        | 16             | ~150 ms | Scale (1.2B)         |
| <b>RTC</b> [107]         | Flow + overlap     | 50        | 10        | 16             | <50 ms* | Interleaved exec.    |
| <b>BID</b> [108]         | Bidirectional      | Variable  | Variable  | 16             | ~100 ms | Dual-end decode      |

\*Effective latency with overlapped computation.

## 8. Bimanual Manipulation with VLAs

Bimanual manipulation is the primary application focus of this review. Prior work has studied bimanual coordination from multiple perspectives: Kreiman *et al.* [16] survey bimanual coordination methods including classical and learning-based approaches, Grannen *et al.* [110] propose a “stabilize to act” framework where one arm stabilizes while the other manipulates, and Chitnis *et al.* [111] learn task schemas for efficient bimanual planning. As shown in the “Bimanual” branch of Figure 1, this section examines how VLAs address the unique challenges of bimanual coordination, organized by coordination strategy (Table 9) and task type (Table 7). Table 8 lists the principal hardware platforms. The action representations and chunking strategies discussed in Section 7 are central to the coordination mechanisms described below. Figure 4 illustrates the ALOHA platform and its bilateral teleoperation interface.



**Figure 4.** The ALOHA bimanual teleoperation platform and representative tasks. A human operator controls two follower arms via leader arms for intuitive demonstration collection. ALOHA and its ACT policy established the standard platform for bimanual VLA research. Figure obtained from Zhao *et al.* [17].

### 8.1. Coordination Strategies

#### 8.1.1. Joint Action Space

The most common approach treats the bimanual system as a single high-dimensional policy. The action chunk  $\mathbf{A}_t \in \mathbb{R}^{H \times (d_L + d_R)}$  encodes both arms jointly, allowing the model to learn implicit coordination.  $\pi_0$  [2] and **RDT-1B** [44] both use this approach, predicting the left and right arm actions

as a concatenated vector. The advantage is simplicity: no explicit coordination mechanism is needed, and the model can learn arbitrary coordination patterns from data. The disadvantage is that the action space is large ( $H \times 16$  dimensions for typical settings), requiring expressive generative models to capture the joint distribution.

Flow matching is well-suited to this approach because it generates the entire action chunk in a single denoising process, naturally preserving inter-arm correlations. The flow field  $\mathbf{v}_\theta$  operates on the full  $H \times (d_L + d_R)$ -dimensional space, learning the joint velocity field that transports noise to coordinated bimanual trajectories. This global denoising preserves correlations between left and right arm motions at every timestep within the chunk.

In contrast, autoregressive generation of a joint action vector must predict left and right arm actions in some sequential order, potentially breaking symmetry. If the model generates left arm actions before right arm actions (or vice versa), the second arm's predictions are conditioned on the first arm's, introducing an artificial asymmetry that may not reflect the actual task structure. While this asymmetry can be mitigated through data augmentation (randomly swapping left and right arm labels), it remains a conceptual limitation of autoregressive bimanual action generation.

### 8.1.2. Independent Policies

An alternative is to train separate policies for each arm, coordinated by a high-level planner. **Hi Robot** [29] decomposes bimanual tasks hierarchically: a high-level VLM generates subtask descriptions for each arm, and separate low-level policies execute them. This approach simplifies each policy's action space but requires the high-level planner to handle coordination timing and conflict avoidance.

Independent policies work well for loosely coupled tasks (*e.g.*, one arm holds an object while the other operates on it) but struggle with tightly coupled tasks (*e.g.*, folding, where both arms must move in concert). The coordination information that joint policies learn implicitly must be provided explicitly through the high-level planner's instructions.

### 8.1.3. Leader-Follower

In leader-follower coordination, one arm (the leader) executes a primary manipulation while the other (the follower) adapts to maintain a constraint (*e.g.*, holding an object stable). This asymmetric decomposition reduces the effective planning complexity and can be encoded in the VLA by conditioning one arm's actions on the other's predicted trajectory. Several bimanual VLA systems implement soft leader-follower coordination implicitly through the joint action space, where the model learns that one arm typically initiates contact while the other provides support.

## 8.2. Contact-Rich Bimanual Tasks

Contact-rich tasks, where both arms simultaneously exert forces on an object, are among the most challenging for VLAs. Examples include inserting a peg with one arm while the other holds the socket, tightening a cap on a bottle held by the other arm, and assembling parts that require precise force alignment.

Contact-rich bimanual performance has been most thoroughly evaluated on box assembly, where one arm holds a box while the other folds flaps.  $\pi_0$  [2] achieves smooth force profiles on this task through its flow-matching action head, avoiding the jerkiness of discrete-action policies. An equally important insight came from **ALOHA (ACT)** [17]: action chunking is critical for contact-rich tasks, as single-step predictions produce oscillatory contact forces while chunked predictions maintain stable contact throughout a manipulation primitive.

Most VLAs operate in position or velocity space without explicit force feedback, inferring contact states from visual cues alone.  $\pi_0$  [2] learns appropriate forces during box assembly purely from visual demonstrations, but vision alone likely breaks down for high-precision force-sensitive operations.  $\pi_0^*$  [4] partially addresses this through RL from autonomous practice, where the robot discovers effective force profiles through trial and error. The difficulty scales with contact points: current VLAs

handle two-point contact well, but multi-point contact and multi-fingered dexterous manipulation remain open frontiers.

### 8.3. Deformable Object Manipulation

Deformable objects (fabric, rope, dough, plastic bags) present distinct challenges for bimanual VLAs: the object state is high-dimensional and partially observable, and manipulation requires coordinated bimanual actions that account for material dynamics.

Laundry folding is the canonical bimanual deformable-object task and, until recently, an unsolved problem. Success rates above 80% on T-shirt folding became possible when  $\pi_0$  [2] combined flow matching with long action chunks, learning the complex bimanual coordination required to pinch, lift, fold, and smooth fabric. The success relies on long action chunks ( $H = 50$ , as analyzed in Section 7.3) that capture the full folding motion as a continuous trajectory, and on the VLM backbone’s ability to visually parse the garment’s configuration.

Building on this,  $\pi_{0.5}$  [3] generalized folding to arbitrary garments in novel homes. The hierarchical architecture decomposes folding into subgoals (e.g., “pick up the left sleeve”, “fold it toward the center”), with the high-level VLM reasoning about garment topology and the low-level policy handling motor execution. Current success rates on fabric folding are measured on a narrow range of garment types; generalization to thin, slippery, or multi-layered fabrics remains undemonstrated.

Beyond fabric, bimanual deformable-object manipulation encompasses rope knotting, dough shaping, and bag opening. The difficulty increases from 1D deformation (rope) through 2D (fabric with self-occlusion) to 3D (dough, clay). Current VLAs address these through large action chunks that capture entire primitives, which works when deformation is predictable but fails for materials with complex dynamics. Integrating tactile sensing to detect material state is a promising direction.

### 8.4. Long-Horizon Bimanual Tasks

Long-horizon tasks require the robot to execute many bimanual primitives in sequence, with the correct ordering determined by task semantics. Table clearing, for example, requires picking up plates, stacking them, wiping the table, and placing items in a bin: a sequence of 10–20 bimanual actions over several minutes.

$\pi_{0.5}$  [3] handles long-horizon tasks through its hierarchical architecture: the high-level VLM maintains a task plan and generates subgoal instructions, while the low-level policy executes each subgoal. The high-level model can re-plan based on visual feedback, recovering from failures or adapting to unexpected states. **Hi Robot** [29] similarly uses hierarchical VLA reasoning for open-ended instructions, decomposing “tidy the desk” into a sequence of specific bimanual actions.

Prior work on LLM-based planning provides the conceptual foundation for long-horizon VLA reasoning. **Code as Policies** [112] uses LLMs to generate executable code that sequences manipulation primitives, while **SayCan** [113] grounds LLM proposals in learned affordance scores, ensuring that only feasible actions are selected. Both approaches separate high-level reasoning from low-level execution, a principle adopted by hierarchical VLAs.

Complementary approaches learn long-horizon skills from unstructured data. **MimicPlay** [114] decomposes human play videos into plan representations, **PlayFusion** [115] acquires skills via diffusion from play data, and Du *et al.* [116] learn policies via text-guided video generation. **Look Before You Leap** [117] uses GPT-4V to preview action consequences, and **Language-Image Reward Models** [118] provide scalable reward signals for RL-based improvement. These planning strategies predate VLAs but are complementary and could be integrated with VLA execution.

Long-horizon bimanual tasks demand robust *error recovery*. Hierarchical VLAs address this naturally: the high-level VLM detects failures and generates recovery subgoals.  $\pi_0^*$  [4] improves recovery through autonomous practice (Section 6.3), learning robust behaviors that pure imitation cannot provide. The temporal extent also poses a computational challenge: a 5-minute task at 50 Hz involves 15,000 control steps (300 chunk-level decisions with  $H = 50$ ), requiring either hierarchical planning or long-context models.

Recent work addresses this limitation by equipping VLAs with explicit *memory* mechanisms. **MEM** [77] introduces *Multi-Scale Embodied Memory*, a system that combines two complementary modalities: a video encoder for short-horizon, image-based memory (which supports in-context adaptation and occlusion recovery over seconds) and a language-based memory that maintains compressed text summaries of semantic events over long horizons (up to 15 minutes). Integrated into the  $\pi_{0.6}$  VLA, MEM achieves state-of-the-art results on tasks such as recipe setup, kitchen cleanup, and grilled cheese preparation, while matching non-memory VLAs on standard dexterous manipulation. Different time scales require different memory representations: dense visual context for recent events and compressed language for long-term semantic state.

Concurrent approaches explore complementary designs. Context-compression methods include **ContextVLA** [79] (amortizing multi-frame context into a single token), **CronusVLA** [119] (learnable temporal feature chunking), **BPP** [120] (conditioning on VLM-detected keyframes), and past-token prediction [121] ( $3\times$  gains at  $10\times$  reduced cost). Retrieval-based methods include **MemoryVLA** [78] (perceptual-cognitive memory bank; +14.6% on Bridge), **SAM2Act** [122] (episodic spatial memory; 86.8% across 18 RL Bench tasks), **MemER** [123] (VLM-guided keyframe retrieval), and **CycleManip** [124] (cost-aware historical sampling for cyclic tasks).

Memory-augmented VLAs enable long-horizon execution (up to 15 minutes), in-context adaptation, and partial observability handling. However, they face two intertwined challenges: *causal confusion*, where the policy learns to copy its own past actions rather than reason about the current state, and *train-inference shift*, where self-generated memory summaries at test time may contain compounding errors. MEM mitigates causal confusion via language compression that discards failed attempts, but the general problem remains unsolved. Additional limitations include computational overhead that scales with history length and information loss from memory compression (semantic events are retained, but fine-grained contact forces are not). Current memory systems are episodic, with no mechanism for accumulating knowledge across deployment sessions.

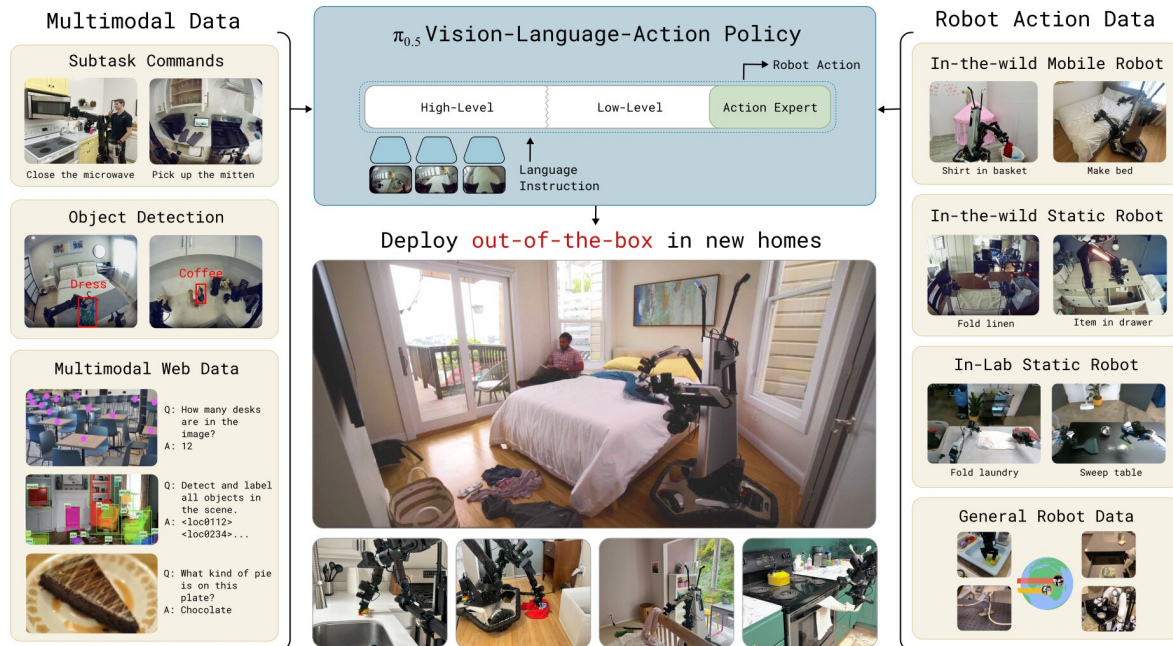
### 8.5. Mobile Bimanual Manipulation

Mobile bimanual systems add navigation to the manipulation challenge. The robot must move to the task location, position itself appropriately, and then perform bimanual manipulation, requiring coordination between the base and both arms.

Whole-body teleoperation and imitation learning for mobile bimanual tasks were first shown by **Mobile ALOHA** [42], which controls the mobile base and two arms simultaneously for tasks such as cooking and furniture assembly, with action chunks covering all degrees of freedom.

$\pi_{0.5}$  [3] deployed VLA-controlled mobile bimanual robots in real homes, where it reached a 98% success rate on household tasks. The hierarchical architecture separates navigation decisions (handled by the high-level VLM) from manipulation execution (handled by the low-level policy), simplifying the learning problem.

*Base-arm coordination* is the additional challenge: the base must position itself so both arms reach target objects. **UMI-on-Legs** [125] extends hardware-agnostic data collection to legged platforms, **BUMBLE** [126] addresses building-wide mobile manipulation, and industrial efforts (**AgiBot World** [127], **Physical Intelligence** [128]) are scaling bimanual fleet data and policy learning. Figure 5 shows  $\pi_{0.5}$  deployed in real homes.



**Figure 5.** Real-world deployment of  $\pi_{0.5}$  in homes. A hierarchical VLA decomposes high-level instructions into subgoals, with 98% success on household tasks such as table clearing and laundry folding. Figure obtained from Black *et al.* [3].

**Table 7.** Approximate success rates (%) on bimanual tasks, as reported in the original publications under varying evaluation conditions. Values across methods are not directly comparable due to differences in task definitions, object sets, and evaluation protocols. Tasks grouped by category.

| Task Category       | Specific Task            | $\pi_0$ | $\pi_0^*$ | RDT-1B | ACT | Diff. Policy |
|---------------------|--------------------------|---------|-----------|--------|-----|--------------|
| <i>Deformable</i>   | Laundry folding          | 80      | 92        | –      | 50  | 35           |
|                     | Towel folding            | 85      | 95        | 60     | 55  | 40           |
| <i>Contact-rich</i> | Box assembly             | 75      | 88        | 55     | 40  | 30           |
|                     | Peg insertion (bimanual) | 70      | 85        | 50     | 45  | 35           |
| <i>Long-horizon</i> | Table busing             | 65      | 80        | –      | 30  | –            |
|                     | Kitchen cleanup          | 60      | 78        | –      | –   | –            |
| <i>Coordination</i> | Object handover          | 90      | 95        | 75     | 70  | 60           |
|                     | Collaborative lift       | 85      | 93        | 70     | 60  | 50           |

**Table 8.** Bimanual hardware platforms used with VLA models. DOF/arm indicates degrees of freedom per arm plus gripper. Cost is approximate at time of introduction.

| Platform          | DOF/arm | Mobile | Cost   |
|-------------------|---------|--------|--------|
| ALOHA [17]        | 6+1     | –      | <\$20K |
| Mobile ALOHA [42] | 6+1     | ✓      | <\$30K |
| Franka Dual       | 7+1     | –      | >\$60K |
| UMI [43]          | N/A*    | –      | <\$5K  |

\*UMI is a data collection interface, not a robot.

**Table 9.** Comparison of bimanual coordination strategies in VLA models.  $d_a$  indicates the effective action dimensionality per step. Tight coupling indicates whether the strategy supports simultaneous force-coordinated bimanual actions.

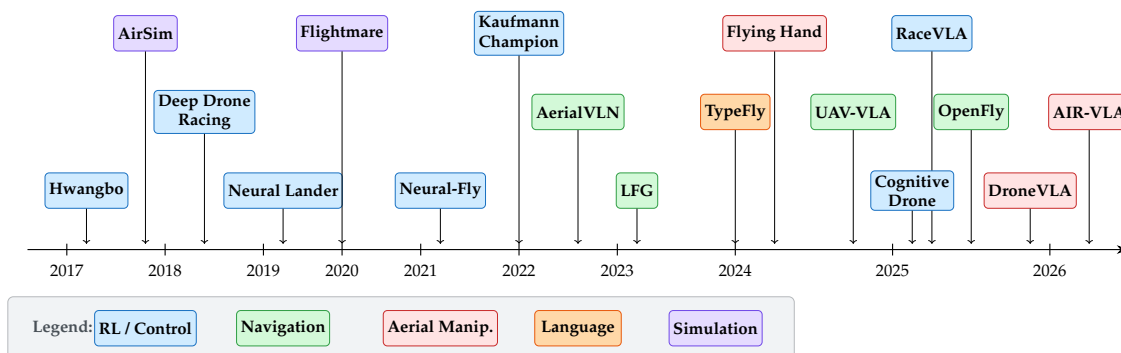
| Strategy        | $d_a$            | Coupling | Methods                |
|-----------------|------------------|----------|------------------------|
| Joint space     | $d_L + d_R$      | ✓        | $\pi_0$ , RDT-1B, ACT  |
| Independent     | $\max(d_L, d_R)$ | –        | Hi Robot               |
| Leader-follower | $d_L$            | Partial  | Custom setups          |
| Hierarchical    | Variable         | ✓        | $\pi_{0.5}$ , Hi Robot |

Flow-based VLAs ( $\pi_0, \pi_0^*$ ) currently lead on most bimanual tasks, with the joint action space approach outperforming decoupled approaches on tightly coupled tasks by preserving inter-arm correlations. For loosely coupled tasks, hierarchical approaches offer interpretability through auditable subgoal decompositions.

The coordination strategies analyzed above, joint action spaces for tightly coupled agents, hierarchical decomposition for complex tasks, leader-follower for asymmetric roles, are not specific to two-armed robots. They apply whenever a VLA must coordinate multiple coupled actuators from shared observations. We now examine a domain where exactly the same coordination problem arises in a different physical setting: unmanned aerial robotics.

## 9. VLA for Unmanned Aerial Robotics and Drones

Section 8 showed how VLAs coordinate two arms through joint action spaces, hierarchical planning, and action chunking. Unmanned aerial robotics faces the same coordination problem in a different physical setting. A single drone must coordinate thrust, attitude, and (optionally) gripper commands; a multi-drone system must coordinate an entire fleet. The VLA machinery (VLM backbone for language grounding, flow matching or diffusion for smooth trajectory generation, action chunking for temporal coherence) applies directly. What changes is the action space: whereas bimanual VLAs generate joint positions or end-effector poses, aerial VLAs must produce velocity commands, waypoints, or low-level thrust-and-torque signals for underactuated platforms operating in three-dimensional space. Latency constraints are stricter ( $\geq 100$  Hz for stable flight versus  $\sim 50$  Hz for manipulation), the observation space often includes GPS, IMU, and depth sensing alongside monocular or stereo vision, and the environment is outdoor, three-dimensional, and wind-affected. Table 10 compares representative methods; Figure 6 charts the key milestones.



**Figure 6.** Timeline of unmanned aerial robotics milestones for learning-based drone control (2017–2026). Colors indicate the research area: RL-based control (blue), vision-language navigation (green), aerial manipulation (red), language-guided planning (orange), and simulation platforms (purple). Early work focused on RL for agile flight and simulators; 2023–2024 saw the emergence of language-guided navigation; 2025–2026 marks the arrival of full VLA systems (CognitiveDrone, RaceVLA, DroneVLA, AIR-VLA) that integrate perception, language, and action generation end-to-end.

## 9.1. VLA-Based Drone Navigation and Control

### 9.1.1. Vision-Language Navigation for UAVs

Vision-language navigation (VLN) requires a drone to reach a goal described in natural language (“fly above the red building”, “turn left at the intersection”) using only visual observations. Although VLN originated in indoor settings, the aerial variant poses distinct challenges: a vastly different visual perspective, a larger action space that includes altitude, and outdoor visual diversity.

The **AerialVLN** benchmark [129] established this task with over 25,000 instruction-trajectory pairs across urban and rural environments, revealing that indoor VLN methods transfer poorly to aerial scenes. A zero-shot alternative, **LFG** [130], sidesteps task-specific training entirely by having an LLM convert language instructions into spatial cost maps that a standard path planner optimizes over. **SkyGPT** [131] pushes this further by coupling vision foundation models with language models for joint scene understanding and decision-making.

The most complete aerial VLA to date is **UAV-VLA** [132], which processes satellite imagery through a VLM backbone and generates full mission plans (waypoints, altitudes, sensor configurations) from natural language. On a 100K-mission dataset it produces plans  $6.5\times$  faster than human operators at comparable quality, showing that VLAs can scale to operational aerial planning beyond single-flight control.

The field is rapidly standardizing. **UAV-VLN** [133] parses instructions into structured sub-goals grounded by a vision model; **OpenFly** [134] provides a large-scale benchmark spanning urban, suburban, rural, and industrial settings; and **CityNavAgent** [135] adds a persistent semantic map that supports city-scale navigation with hierarchical planning.

### 9.1.2. End-to-End Learned Flight Control

End-to-end approaches map raw sensor observations directly to flight commands, bypassing the traditional perception-planning-control pipeline. The feasibility of this approach was established early: a single neural network trained with RL [136] can map quadrotor state directly to motor commands, stabilizing the vehicle even when thrown upside-down at 5 m/s, with policy evaluation taking only  $7\ \mu\text{s}$  per step (Figure 7); this proves that learned policies can replace hand-designed cascaded PID controllers for agile flight.



**Figure 7.** An RL-trained quadrotor recovering from an inverted throw at 5 m/s. The policy maps state to motor commands at  $7\ \mu\text{s}$  per step, establishing the viability of learned end-to-end drone control. Figure obtained from Hwangbo *et al.* [136].

Subsequent RL-based systems pushed the performance frontier to superhuman levels. The landmark result is an autonomous racing policy [137] that processes onboard vision and IMU at 100 Hz and defeated world-champion human pilots. A complementary approach, neural residual dynamics models [138], preserves the nominal controller but learns the gap between the physics model and reality, cutting trajectory tracking error by 40–60%.

Two recent systems bring the VLA framework directly to drones. **CognitiveDrone** [9], trained on 8,000+ simulated trajectories, generates real-time 4D actions ( $x, y, z, \text{yaw}$ ) from first-person imagery and text instructions. Its R1 variant adds VLM-based chain-of-thought reasoning before acting, which lifts the success rate to 77.2%, a 30% gain that demonstrates the value of deliberation for aerial cognitive tasks. **RaceVLA** [139] trains on expert pilot demonstrations annotated with language (“aggressive apex cutting”, “conservative trajectory”) and produces stylistically diverse racing trajectories, going beyond time-optimal control to capture human-interpretable flight behavior.

World models are also gaining traction for aerial control. **Dream to Fly** [140] learns a latent dynamics model from visual observations and plans by simulating future trajectories in the learned space, reducing real-world data needs by an order of magnitude compared to model-free RL (see Section 11). Robustness to real-world disturbances remains a key gap: **Neural-Fly** [141] addresses this through rapid online adaptation, maintaining stable aggressive flight in winds exceeding 12 m/s by learning a wind-invariant representation from just a few flight segments. Diffusion-based policies, originally developed for manipulation (Section 5), are also being applied to generate smooth, multi-modal drone trajectories.

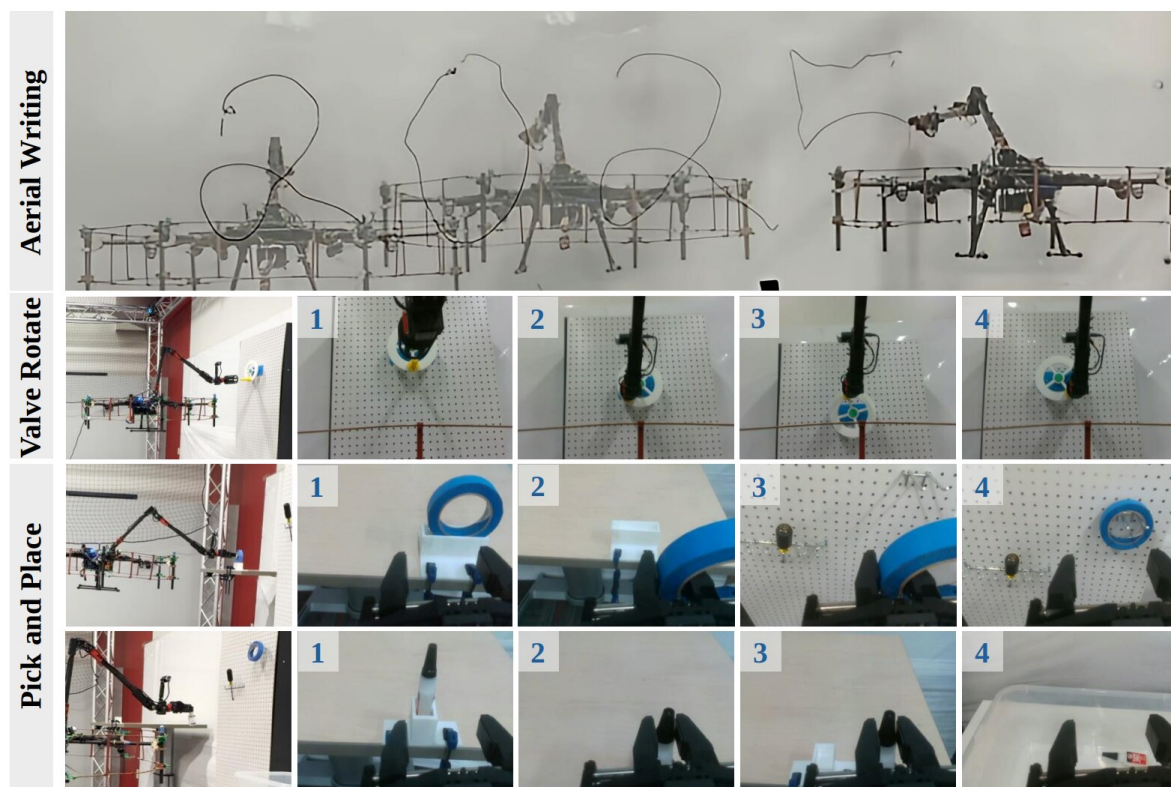
## 9.2. Aerial Manipulation

Aerial manipulation, where drones grasp, transport, and interact with objects, directly inherits the coordination challenges discussed in the bimanual context (Section 8). A drone performing aerial grasping must simultaneously stabilize its flight while executing precise gripper motions, a challenge analogous to bimanual base-arm coordination (Section 8.5).

### 9.2.1. Grasping and Payload Transport

RL-based aerial grasping [142] first demonstrated that a single policy can coordinate flight stabilization with gripper control, learning to compensate for payload-induced dynamics shifts that do not arise in ground-based manipulation. Two recent systems bring the full VLA pipeline to this problem. **DroneVLA** [10] integrates open-vocabulary object detection (Grounding DINO), gripper pose estimation (MediaPipe), and visual servoing into a language-commanded retrieval system: given “pick up the red box and deliver it to the table”, a VLM decomposes the instruction into manipulation sub-goals executed in sequence. **AIR-VLA** [11] takes an end-to-end approach, mapping camera images and language directly to joint flight-and-gripper actions at 20 Hz, with safety constraints (workspace limits, obstacle avoidance) baked in as differentiable penalties during training.

A particularly relevant platform is **Flying Hand** [12], a fully-actuated hexarotor with a 4-DOF arm that formulates control in the end-effector frame, decoupling manipulation precision from flight stabilization (Figure 8). Its imitation learning policy uses ACT (Section 7) to perform writing, peg-in-hole insertion, and pick-and-place, directly demonstrating that action chunking transfers from bimanual to aerial manipulation. The connection to bimanual coordination becomes explicit in an aerial harvesting system [143] where a dual-arm drone picks avocados: one arm stabilizes the branch while the other cuts the fruit, a leader-follower strategy identical to those analyzed in Section 8.1.



**Figure 8.** Flying Hand: a fully-actuated hexarotor with a 4-DOF arm performing writing, peg-in-hole, and pick-and-place via ACT, demonstrating that action chunking transfers from manipulation to aerial systems. Figure obtained from Pan *et al.* [12].

These platforms combine flight commands with gripper commands, yielding high-dimensional action spaces that benefit from the same chunking strategies used for bimanual VLAs. Aerial inspection tasks (close-proximity structure navigation, contact-based measurement) further build on the contact-rich manipulation insights from Section 8.2. A significant gap remains, however: all current aerial manipulation VLAs have been tested only on simplified pick-and-place tasks with lightweight objects. Precision manipulation under wind disturbances and with heavy or awkward payloads has not been demonstrated.

### 9.3. Language-Guided Drone Missions

#### 9.3.1. Natural Language to Flight Plans

The same Code-as-Policies idea used for manipulation [112] (Section 10.2) extends naturally to drones: prompting an LLM with a drone API description lets it convert “survey the perimeter at 20 meters” into executable waypoint commands with no task-specific training. **AeroAgent** [144] adds safety awareness to this approach, decomposing complex missions into atomic flight actions while respecting no-fly zones and altitude limits. A latency bottleneck remains, however: free-form code generation is slow. **TypeFly** [145] addresses this by constraining the LLM to output programs in MiniSpec, a minimal drone-specific language with primitives like takeoff, move, rotate, and sense. The restricted grammar cuts generation latency below 500 ms, an order of magnitude faster than unconstrained approaches, making real-time mission replanning practical.

#### 9.3.2. Interactive and Corrective Language Control

Real-time language correction during flight (“go higher”, “move left”, “stop”) requires low-latency VLA inference. This setting parallels the interactive language control studied for manipulation [146] but with stricter latency requirements due to flight dynamics. Current approaches use lightweight VLM encoders or pre-computed language embeddings to minimize inference overhead.

#### 9.4. Multi-Agent Aerial Systems

Multi-drone coordination with VLAs exhibits direct structural parallels to bimanual coordination (Section 8.1). The joint action space approach for bimanual VLAs, where a single model generates actions for both arms simultaneously, naturally extends to multi-drone systems where a centralized policy generates waypoints for all drones in the swarm.

Multi-drone coordination mirrors the three bimanual strategies from Section 8.1. Centralized policies generate joint actions for all drones but face the same dimensionality scaling as bimanual joint action spaces. Decentralized approaches reduce per-agent complexity but require explicit coordination. Hierarchical approaches, where a high-level VLM assigns subgoals to individual drones (**Hi Robot** [29],  $\pi_{0.5}$  [3]), offer the best scalability. **Decentralized MARL** [147] trains swarm policies where each drone outputs velocity commands from local observations, scaling to 10+ drones. Graph neural network architectures that model inter-drone communication provide a natural framework for swarm VLAs. Integrating VLM-based task assignment with MARL execution is a promising direction. However, no multi-drone VLA has been demonstrated on physical hardware; all results remain simulation-only.

#### 9.5. UAV-UGV Collaborative Systems

Heterogeneous UAV-UGV systems exploit complementary capabilities: drones provide aerial survey while ground robots perform manipulation. VLMs provide a natural coordination interface via language-based task allocation. Cross-embodiment VLA pre-training (Section 10.4) is directly applicable; **Octo** [6] and **Octo 2.0** [148] already span manipulation, navigation, and locomotion embodiments. The coordination challenge mirrors bimanual leader-follower strategies (Section 8.1): one agent (typically the drone) provides context while the other (the ground robot) executes manipulation. No published system yet deploys a shared VLA policy across both a UAV and a UGV in a single mission, making this an open research direction (Section 12.3).

#### 9.6. Sim-to-Real Transfer for Aerial VLAs

Simulation is especially important for aerial VLAs because real-world drone data collection is expensive, risky, and constrained by regulations.

##### 9.6.1. Simulation Environments

Two simulators dominate aerial VLA research. **AirSim** [45], built on Unreal Engine, provides photorealistic rendering with accurate flight dynamics and a rich sensor API (cameras, IMU, GPS, LiDAR). **Flightmare** [46] takes a different approach, decoupling rendering from physics to reach  $\sim 200\times$  real-time speeds for large-scale parallel RL training. Additional platforms (**RotorS** [149], **Isaac Sim**, **Gazebo**) serve complementary fidelity and scale requirements.

Large-scale datasets fill the gap between simulated and real-world training data. **TartanAir** [150] spans hundreds of scenes with diverse weather and lighting; **Mid-Air** [151] focuses on low-altitude flights (1–20 m) with stereo images, depth, and semantic labels.

##### 9.6.2. Domain Adaptation and Reality Gap

The sim-to-real gap for aerial systems involves both visual and dynamics discrepancies. *Visual domain randomization* during training improves transfer of vision-based policies by exposing the model to varied textures, lighting, and weather conditions. *Dynamics randomization* varies mass, inertia, drag, and motor characteristics to produce policies robust to the physical reality gap. These techniques parallel the sim-to-real methods used for manipulation VLAs (Section 11), with the additional challenge that aerodynamic effects (ground effect, wind gusts, rotor wash) are difficult to simulate accurately.

**Table 10.** Representative VLA and learning-based methods for unmanned aerial robotics. Action type indicates the output space of the learned policy. Sim indicates whether the method uses simulation for training.

| Method                                       | Task             | Approach           | Action Type           | Sim | Year | Highlights                              |
|--|------------------|--------------------|-----------------------|-----|------|---|
| <i>Navigation and Control</i>                |                  |                    |                       |     |      |   |
| AerialVLN [129]                              | VL navigation    | VLN baseline       | Waypoints             | ✓   | 2023 | First outdoor aerial VLN benchmark      |
| LFG [130]                                    | Language nav.    | LLM → cost map     | Waypoints             | –   | 2023 | Zero-shot LLM-guided navigation         |
| UAV-VLA [132]                                | Mission gen.     | VLA (sat. imagery) | Waypoints             | –   | 2025 | 6.5× faster than human; 100K missions   |
| UAV-VLN [133]                                | VL navigation    | LLM + vision       | Waypoints             | ✓   | 2025 | End-to-end VLN with LLM parsing         |
| OpenFly [134]                                | VLN benchmark    | Toolchain          | Waypoints             | ✓   | 2025 | Large-scale aerial VLN benchmark        |
| CityNavAgent [135]                           | City-scale nav.  | Hierarchical VLN   | Waypoints             | ✓   | 2025 | Semantic planning + global memory       |
| Hwangbo <i>et al.</i> [136]                  | Stabilization    | RL                 | Motor cmds            | ✓   | 2017 | 7 $\mu$ s inference; thrown recovery    |
| Kaufmann <i>et al.</i> [137]                 | Drone racing     | RL                 | Motor cmds            | ✓   | 2023 | Superhuman agile flight                 |
| CognitiveDrone [9]                           | Cognitive tasks  | VLA                | 4D ( $x, y, z, yaw$ ) | ✓   | 2025 | 77.2% success with VLM reasoning        |
| RaceVLA [139]                                | Drone racing     | VLA                | Velocity cmds         | ✓   | 2025 | Human-like racing behavior              |
| Neural-Fly [141]                             | Agile flight     | Adaptive NN        | Motor cmds            | –   | 2022 | Online adaptation in strong winds       |
| Dream to Fly [140]                           | Vision flight    | Model-based RL     | Velocity cmds         | ✓   | 2025 | Learned world model for planning        |
| <i>Aerial Manipulation</i>                   |                  |                    |                       |     |      |   |
| Zhang <i>et al.</i> [142]                    | Aerial grasp     | RL                 | Thrust + grip         | ✓   | 2019 | Flight-grasp coordination               |
| DroneVLA [10]                                | Object retrieval | VLA + servoing     | EE pose + grip        | –   | 2026 | Language-commanded aerial manipulation  |
| AIR-VLA [11]                                 | Aerial manip.    | End-to-end VLA     | Flight + grip         | –   | 2026 | Safety-constrained; 20 Hz inference     |
| Flying Hand [12]                             | Dexterous manip. | ACT + MPC          | 6-DOF + 4-DOF arm     | –   | 2025 | Hexarotor; writing, peg-in-hole         |
| Aerial Bimanual [143]                        | Harvesting       | Dual-arm aerial    | Dual-arm cmds         | –   | 2024 | Bimanual aerial manipulation            |
| <i>Language-Guided Missions</i>              |                  |                    |                       |     |      |   |
| AeroAgent [144]                              | Mission plan     | LLM agent          | API calls             | –   | 2025 | LLM mission decomposition               |
| TypeFly [145]                                | Mission plan     | LLM → MiniSpec     | API calls             | –   | 2024 | Low-latency program generation          |
| <i>Multi-Agent</i>                           |                  |                    |                       |     |      |   |
| MARL Swarms [147]                            | Formation        | Decentralized MARL | Velocity cmds         | ✓   | 2021 | Scalable to 10+ drones                  |
| <i>Simulation, Datasets, and Sim-to-Real</i> |                  |                    |                       |     |      |   |
| AirSim [45]                                  | Sim platform     | UE4 rendering      | Various               | ✓   | 2018 | Photorealistic drone sim                |
| Flightmare [46]                              | Sim platform     | Parallel RL        | Various               | ✓   | 2021 | 200× real-time training                 |
| TartanAir [150]                              | Dataset          | Multi-modal        | –                     | ✓   | 2020 | Diverse visual conditions; SLAM focus   |
| Mid-Air [151]                                | Dataset          | Multi-modal        | –                     | ✓   | 2019 | Low-altitude flights; depth + semantics |

The aerial landscape shows rapid progress: 2025–2026 has seen a surge of end-to-end aerial VLA systems (UAV-VLA, CognitiveDrone, DroneVLA, AIR-VLA, Flying Hand) that directly map observations and language to flight actions. The bridging of bimanual coordination with aerial manipulation [12,143] suggests bidirectional technical transfer. Research directions for advancing this convergence appear in Section 12.

## 10. Language Grounding, Reasoning, and Generalization

VLAs condition on natural-language instructions, inheriting the semantic understanding of pre-trained VLMs. This section examines language grounding, hierarchical reasoning, open-ended instruction following, and cross-embodiment generalization, mechanisms applicable to both manipulation and aerial domains.

### 10.1. Language-Conditioned Policies

The roots of language-conditioned policy learning trace back to **Language-conditioned IL** [32], which established multi-task learning from language instructions, and **BC-Z** [152], which scaled it to zero-shot generalization. VLAs extend this lineage by processing language and image tokens through shared Transformer layers, supporting deep cross-modal reasoning. Emergent grounding (following novel phrasings and generalizing to unseen objects) was first observed in **RT-2** [1]. In contrast,  $\pi_0$  [2] conditions the flow-matching head on VLM hidden representations, grounding abstract instructions in continuous motor behaviors rather than discrete tokens. A persistent weakness is brittle instruction parsing: minor rephrasing or typographical errors can cause large performance drops.

### 10.2. Hierarchical Reasoning

Complex instructions require decomposition into executable subgoals. **Hi Robot** [29] introduces a hierarchical VLA architecture where a high-level “reasoner” VLM processes the user’s open-ended instruction and the current visual observation to generate a specific subgoal instruction. A low-level “executor” VLA then carries out the subgoal. This decomposition allows Hi Robot to follow complex instructions such as “make me a sandwich” by generating subgoals such as “open the bread bag”, “pick up two slices”, “place cheese between them”.

$\pi_{0.5}$  [3] implements a similar hierarchy, with the high-level model additionally maintaining a task state representation that tracks progress through multi-step tasks. The high-level model can detect when a subgoal has failed and re-plan, providing robustness to execution errors.

Prior work established hierarchical planning principles adopted by VLAs: **SayCan** [113] grounds LLM plans in affordance scores, **Code as Policies** [112] generates executable code, and **VoxPoser** [153] produces 3D value maps. These affordance-grounding principles are incorporated in hierarchical VLAs such as Hi Robot and  $\pi_{0.5}$ .

The language channel between high-level planner and low-level executor determines control granularity. Natural language subgoals (**Hi Robot** [29]) are flexible but may be ambiguous; alternatives include code-based specifications [112] (precise but brittle) and goal images (rich but expensive to generate). Natural language is currently favored for its compatibility with VLM backbones.

### 10.3. Open-Ended Instruction Following

Open-ended instruction following tests generalization beyond the training distribution. **RT-2** [1] showed VLM pre-training enables following instructions involving novel objects absent from robot data. Related work includes **Manipulate-Anything** [154,155] (detailed instruction following), **Interactive Language** [146] (real-time streaming corrections), and **Chain-of-Thought Predictive Control** [156] (reasoning-guided action generation).

VLM backbones bridge the gap between simple human commands (“fold the towel”) and the motor detail required for execution;  $\pi_0$  [2] and  $\pi_{0.5}$  [3] operate effectively with natural instructions. A persistent limitation is the lack of systematic evaluation: most results use hand-picked instruction sets, and current VLAs cannot reliably detect ambiguous or contradictory commands.

### 10.4. Cross-Embodiment Transfer

A key promise of VLAs is cross-embodiment generalization: a policy trained on data from multiple robots can be deployed on a new robot with minimal fine-tuning. The evidence is now substantial. **Octo** [6] pre-trains on OXE data spanning 22 embodiments and transfers to unseen robots with a few hundred fine-tuning demonstrations, while **OpenVLA** [5] confirms that OXE pre-training improves performance even on embodiments absent from the pre-training set.

Cross-embodiment transfer is critical for dual-arm systems because bimanual demonstration data is scarce (as discussed in Section 6.4). A VLA pre-trained on diverse single-arm data can transfer visual and semantic representations to a bimanual system, even though the action space differs. The strongest evidence comes from  $\pi_0$  [2], which pre-trains on a mixture of single-arm and bimanual data and finds that single-arm data improves bimanual performance through shared visual representations.

The transition of VLA technology from research to industrial deployment is exemplified by **Xiaomi-Robotics-0** [157], which trains on data from multiple Xiaomi robot platforms with real-time execution optimizations for consumer hardware.

Cross-embodiment transfer for bimanual systems requires handling different arm configurations. Approaches include action space normalization ( $\pi_0$  [2] maps to a common end-effector format), embodiment-specific projection layers (**Octo** [6], extended to navigation and locomotion in **Octo 2.0** [148]), and language-based action hierarchies (**RT-H** [158]).

### 10.5. Zero-Shot and Few-Shot Generalization

Zero-shot generalization (performing tasks with no task-specific training data) remains challenging for VLAs but is an active research frontier. **OK-Robot** [159] combines a VLM for object detection with a pre-trained manipulation primitive to achieve zero-shot pick-and-place in novel environments. While not a full VLA, OK-Robot demonstrates the potential of VLM-based perception for zero-shot manipulation.

**Robot Utility Models** [160] train VLAs as general-purpose “utilities” that can perform a broad range of manipulation tasks from language instructions, approaching zero-shot capability for common

manipulation primitives. The remaining gap to true zero-shot bimanual manipulation is substantial, as bimanual coordination patterns are difficult to infer from language alone without motor experience.

A limitation of current cross-embodiment claims is that transfer is evaluated after fine-tuning without controlling for data quantity; rigorous ablations separating pre-training benefit from fine-tuning benefit are needed.

Table 11 summarizes the generalization capabilities observed across VLA methods, distinguishing between environment, object, instruction, and embodiment generalization.

**Table 11.** Generalization capabilities of VLA models across four dimensions. Strong/Partial/Weak indicate the degree of demonstrated generalization.

| Method          | Env.    | Obj.    | Instr.  | Embod.  |
|-----------------|---------|---------|---------|---------|
| RT-1 [60]       | Weak    | Weak    | Weak    | –       |
| RT-2 [1]        | Partial | Strong  | Strong  | –       |
| OpenVLA [5]     | Partial | Partial | Partial | Partial |
| Octo [6]        | Partial | Partial | Partial | Strong  |
| $\pi_0$ [2]     | Partial | Strong  | Strong  | Partial |
| $\pi_{0.5}$ [3] | Strong  | Strong  | Strong  | Partial |
| Hi Robot [29]   | Partial | Partial | Strong  | –       |

Tables 11 and 12 summarize the generalization and language grounding landscape across VLA methods. Language and cross-embodiment capabilities do not exist in isolation; they interact with visual representations, safety requirements, and deployment constraints, which we address in Section 11.

**Table 12.** Comparison of language grounding and reasoning capabilities in VLA models. Novel instr. indicates generalization to unseen instruction phrasings. Novel obj. indicates generalization to unseen object categories.

| Method          | Hierarchical | Novel Instr. | Novel Obj. | Open-Ended | Cross-Embod. | Zero-Shot |
|-----------------|--------------|--------------|------------|------------|--------------|-----------|
| RT-1 [60]       | –            | Limited      | Limited    | –          | –            | –         |
| RT-2 [1]        | –            | ✓            | ✓          | Partial    | –            | Partial   |
| OpenVLA [5]     | –            | ✓            | ✓          | –          | ✓            | –         |
| $\pi_0$ [2]     | –            | ✓            | ✓          | –          | ✓            | –         |
| $\pi_{0.5}$ [3] | ✓            | ✓            | ✓          | ✓          | ✓            | Partial   |
| Hi Robot [29]   | ✓            | ✓            | ✓          | ✓          | –            | –         |
| SayCan [113]    | ✓            | ✓            | –          | ✓          | –            | –         |

## 11. Cross-Cutting Concerns

Several concerns cut across all VLA architectures and application domains. This section addresses visual representation learning, world models and future state prediction, safety, sim-to-real transfer, and human-robot interaction. Table 13 compares these capabilities across representative VLA methods.

### 11.1. Visual Representation Learning

The choice of visual encoder significantly impacts VLA performance. Three approaches dominate.

**Pre-trained VLM encoders** (e.g., SigLIP in PaLiGemma [25], ViT in CLIP [21]) provide rich semantic features pre-trained on web-scale data. These encoders excel at object recognition and scene understanding but may lack fine-grained spatial information needed for precise manipulation. Spatial structure may be as important as semantic richness for manipulation-oriented visual encoders: **Transporter Networks** [161] achieve strong rearrangement performance using equivariant spatial representations learned without large-scale pre-training.

**Robot-specific visual representations** offer an alternative to generic VLM encoders. **R3M** [162], pre-trained on robot video data using time-contrastive and language-aligned objectives, captures temporal dynamics and manipulation-relevant features that generic encoders miss. Offline data paired with crowd-sourced annotation also yields transferable representations [163]. A large-scale comparison

by **Cortex** [164] found that representations trained on diverse egocentric video outperform those from static image classification, while **SPA** [165] adds explicit 3D spatial-awareness to improve embodied policy learning. Several VLAs use R3M or similar robot-specific encoders alongside VLM encoders, processing images through both pathways.

**Multi-view fusion** is critical for bimanual manipulation, where a single camera may not capture both arms and the workspace simultaneously. Most bimanual VLAs ( $\pi_0$  [2], **RDT-1B** [44]) use multiple camera views (typically a wrist camera on each arm plus one or more third-person cameras) and fuse the resulting tokens within the Transformer backbone.

Multi-view fusion approaches range from early concatenation ( $\pi_0$  [2], **Octo** [6]) to late fusion and learned view selection. Wrist cameras are indispensable for bimanual setups, capturing fine-grained contact information that third-person cameras miss. Multi-frame visual context [77,79,119] extends temporal scope beyond the current observation. A limitation across all three approaches is that no principled method exists for selecting which visual encoder or fusion strategy best suits a given task; current practice relies on empirical trial-and-error, and the relative contribution of semantic versus spatial features to bimanual coordination remains poorly understood.

### 11.2. Safety

Safety is paramount for bimanual systems operating near humans. VLA safety concerns include:

**Action bounds and rate limiting:** VLA outputs are typically clipped to safe action ranges and rate-limited to prevent high-velocity motions. Two-arm coordination introduces an additional collision-avoidance constraint between the arms that is not inherently captured by the VLA.

**Out-of-distribution detection:** Some VLAs use confidence-based filtering, halting when action head uncertainty (estimated from denoising variance or velocity field norms) exceeds a threshold.

**Collision avoidance:** Bimanual systems face self-collision risk between arms. Post-hoc safety layers that project actions to collision-free trajectories add latency but provide guarantees; learning collision avoidance from demonstrations is an alternative that may not generalize to novel configurations.

**Human-in-the-loop correction:** For deployment in homes (e.g.,  $\pi_{0.5}$  [3]), the ability for humans to intervene and correct the robot is essential. VLAs that accept real-time language feedback can be redirected mid-task, providing a natural correction mechanism.

### 11.3. Sim-to-Real Transfer

Simulation provides scalable, safe data generation, but the reality gap (differences between simulated and real visual appearances, physics, and dynamics) limits direct transfer.

**SIMPLER** [53] provides simulation environments calibrated to match real-world VLA evaluation setups, so that VLAs can be evaluated without physical hardware. The correlation between simulated and real-world success rates validates simulation as a development tool for VLAs.

The reality gap is acute for bimanual manipulation because contact dynamics (friction, deformation, compliance) are difficult to simulate accurately. Current bimanual VLAs ( $\pi_0$  [2],  $\pi_0^*$  [4]) rely primarily on real-world demonstrations and practice, with simulation playing a secondary role.

Strategies for closing this gap include **domain randomization** (varying visual and physical parameters), **system identification** (calibrating simulation to match the real robot), and generative approaches such as **Gen2Act** [166] (human video demonstrations) and **Track2Act** [167] (point tracks from internet videos). **Hybrid training** (simulation pre-training plus real fine-tuning) works well for single-arm VLAs but remains underexplored for bimanual systems, where higher action dimensionality makes sim-to-real alignment harder.

### 11.4. World Models and Future State Prediction

A complementary approach to reactive VLA policies is to equip robots with *world models* that predict future states, whether as visual frames, latent representations, or explicit physical quantities, before committing to actions. This “predict-then-act” approach offers several advantages for bimanual manipulation: it allows look-ahead planning for multi-step coordination, provides a mechanism for

evaluating action consequences before execution, and can generate synthetic training data to alleviate the data scarcity problem.

The GigaBrain family illustrates the rapid maturation of world model-powered VLAs. **GigaBrain-0** [80] first used generative models to produce synthetic robot data. Its successor, **GigaBrain-0.5M\*** [51], added RAMP (RL via World Model-conditioned Policy), yielding  $\sim 30\%$  improvement on bimanual tasks. **GigaWorld-0** [83] completes the picture with a unified framework combining video generation and 3D modeling (Gaussian Splatting) as a scalable data engine.

A more radical approach formulates control as video generation. **Rhoda AI (DVA)** [81] uses a causal video model pre-trained on web-scale video, with an inverse dynamics model translating predicted frames to actions (10–20 hours of robot data). **VPP** [168] learns implicit inverse dynamics via video diffusion (+18.6% on Calvin ABC-D). **ViPRA** [169] learns from actionless videos at 22 Hz, and **Mimic-Video** [170] achieves  $10\times$  sample efficiency over standard VLAs.

Other approaches operate on optical flow and latent representations. **FOFPred** [171] achieves 68.6% on bimanual tasks via language-driven flow prediction. **V-JEPA 2** [172] provides a self-supervised world model (1M+ hours of video; 65–80% zero-shot success). **WorldVLA** [82] jointly generates actions and future frames, **UP-VLA** [173] uses next-frame prediction for implicit physics, and **NVIDIA Cosmos** [174] provides open world foundation models trained on 20M+ hours of data.

The predict-then-act approach offers four advantages: *data efficiency* (DVA requires only 10–20 hours of robot data; Mimic-Video achieves  $10\times$  sample efficiency), *interpretability* (predicted frames can be visualised), *planning* (look-ahead evaluation before committing), and *synthetic data generation* (GigaWorld-0 and Cosmos produce unlimited training data).

However, fundamental limitations remain. *Prediction accuracy degrades* as compounding errors in autoregressive video generation make long-horizon forecasts unreliable, especially for contact-rich bimanual tasks with rapid state changes. *Inverse dynamics accuracy* suffers from the additional error of translating predicted video back to precise actions. *Computational cost* conflicts with real-time control budgets (ViPRA runs at only 22 Hz). Video models can *hallucinate* plausible but incorrect states after occlusions, and current world models lack *haptic grounding* for force and tactile signals.

### 11.5. Human-Robot Interaction

VLAs facilitate more natural human-robot interaction through language. A user can instruct a bimanual robot in natural language, observe its behavior, and provide corrections or new instructions in real time.

The most complete HRI demonstration to date comes from  $\pi_{0.5}$  [3], where users gave verbal instructions to a bimanual robot in home environments, the robot executed them, and users could redirect it as needed. The hierarchical architecture allows the robot to ask clarifying questions through the high-level VLM when instructions are ambiguous.

**PaLM-E** [24] and **PIVOT** [175] show that VLMs can engage in dialogue about the physical world and elicit actionable knowledge through visual prompting. However, HRI evaluation for VLAs remains qualitative; no standardized metrics exist for interaction quality or correction latency in VLA-based bimanual systems.

### 11.6. Scalability and Deployment

Deploying VLAs on bimanual systems in real-world settings introduces engineering challenges beyond model performance. **Compute requirements** are substantial: a 3B-parameter VLA running flow matching with  $K = 10$  steps requires a high-end GPU (A100 or equivalent) for real-time bimanual control. Edge deployment on embedded GPUs is not yet practical for full-size VLAs, motivating the efficient architectures discussed in Section 5.4.

**Communication latency** between the VLA compute server and the robot controller adds to the end-to-end control delay. With dual-arm systems running at 50 Hz, the total loop delay (image capture, network transfer, VLA inference, action transfer, motor execution) must remain below 20 ms per

step. Action chunking mitigates this by amortizing the VLA inference over  $H$  steps, but introduces a minimum reaction latency of one chunk period.

A gap remains between research demonstrations and reliable deployment: most results use controlled laboratory conditions, and long-term reliability metrics are absent. Reproducibility is limited by leading systems' reliance on proprietary data. **Xiaomi-Robotics-0** [157] represents industrial VLA deployment with custom hardware accelerators for low-latency bimanual control. **TidyBot** [176] demonstrates LLM-powered household robotics, and **ManiWAV** [177] shows that auditory feedback complements vision for contact-rich tasks.

The interplay among these architectural, training, and deployment considerations shapes the current state of the art, which we synthesize next.

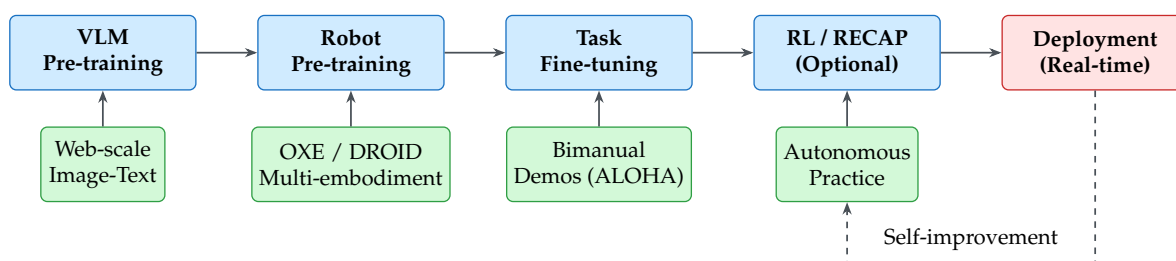
**Table 13.** Cross-cutting capabilities of VLA models. Multi-view indicates support for multiple camera inputs. Safety indicates explicit safety mechanisms. Sim-to-Real indicates simulation-to-real transfer capability.

| Method          | Visual Encoder | Multi-View | Safety      | Sim-to-Real | HRI     | Proprioception |
|-----------------|----------------|------------|-------------|-------------|---------|----------------|
| RT-1 [60]       | EfficientNet   | –          | Basic       | –           | –       | –              |
| RT-2 [1]        | ViT (PaLI-X)   | –          | Basic       | –           | Partial | –              |
| OpenVLA [5]     | DINOv2+SigLIP  | –          | –           | –           | –       | –              |
| $\pi_0$ [2]     | SigLIP         | ✓          | Rate limit  | –           | –       | ✓              |
| $\pi_{0.5}$ [3] | SigLIP         | ✓          | Multi-layer | –           | ✓       | ✓              |
| $\pi_0^*$ [4]   | SigLIP         | ✓          | Rate limit  | –           | –       | ✓              |
| RDT-1B [44]     | SigLIP         | ✓          | Basic       | –           | –       | ✓              |
| Octo [6]        | Custom ViT     | ✓          | –           | ✓           | –       | –              |

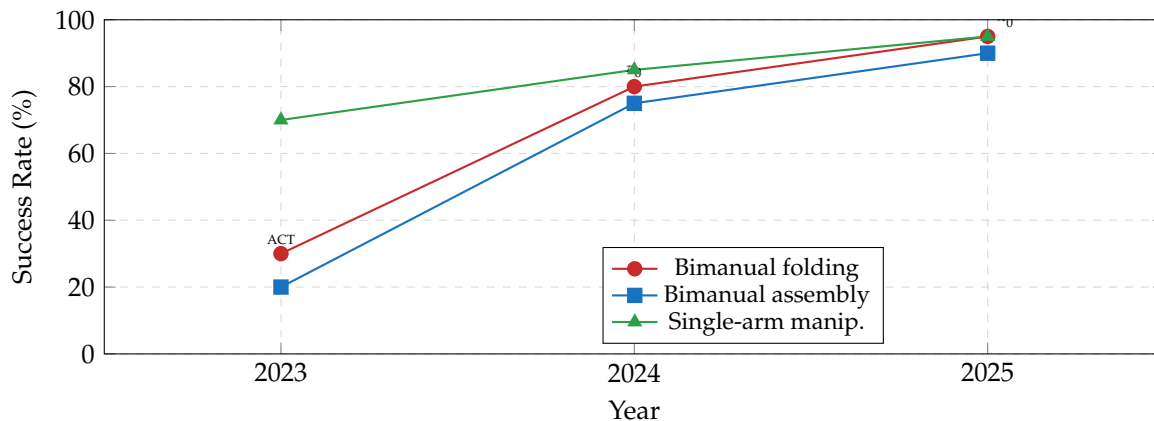
## 12. Discussion and Conclusions

### 12.1. State of the Art Performance

Synthesizing the results presented across Sections 5–11, in our assessment, the state of the art in VLA-based bimanual manipulation and unmanned aerial robotics can be characterized along several dimensions. Figure 10 charts the rapid progress from 2023 to 2025: bimanual folding success rose from ~30% (ACT) to over 90% ( $\pi_0^*$ ), narrowing the gap with single-arm performance.



**Figure 9.** The VLA training and deployment pipeline for bimanual manipulation. Training proceeds through four phases: VLM pre-training on web data, robot pre-training on cross-embodiment datasets, task fine-tuning on bimanual demonstrations, and optional reinforcement learning from autonomous practice. The RL phase creates a self-improvement loop where the deployed policy generates additional training data.



**Figure 10.** Approximate evolution of VLA performance on bimanual manipulation tasks (2023–2025). Values are approximate trend values synthesized by the authors from reported results across different evaluation setups and task definitions; they illustrate general trends rather than exact comparable benchmarks. Bimanual task success rates have improved dramatically, from ~30% with early methods such as ACT [17] to >90% with  $\pi_0^*$  [4]. The gap between bimanual and single-arm performance has narrowed but persists for the most dexterous tasks.

**Architecture.** As reflected in Tables 4 and 7, flow-based VLAs, led by  $\pi_0$  [2] and its successors  $\pi_{0.5}$  [3] and  $\pi_0^*$  [4], currently achieve the strongest bimanual manipulation performance. The flow-matching action head generates smooth, high-dimensional action chunks without quantization, and the iterative denoising process captures the multimodal coordination patterns inherent in bimanual tasks. Diffusion-based models, particularly **RDT-1B** [44], demonstrate that scale improves performance but trail flow-based approaches in inference efficiency. Autoregressive VLAs (**OpenVLA** [5], **RT-2** [1]) provide the simplest integration with VLM pre-training but are limited by discretization for bimanual action spaces.

**Training.** The three-stage recipe (VLM initialization, cross-embodiment pre-training, task-specific fine-tuning) is now standard. The addition of reinforcement learning from autonomous practice (**RECAP** [4]) represents the most impactful recent advance; it allows VLAs to surpass demonstration quality by 10–40% on bimanual tasks. Co-training with diverse data during fine-tuning consistently improves performance and robustness.

**Action representation.** Action chunking with  $H = 50$  steps is the dominant choice for bimanual VLAs, providing the temporal coherence needed for coordinated two-arm motions. Learned action tokenization (**FAST** [8]) narrows the gap between autoregressive and continuous approaches. Real-time execution techniques (**RTC** [107]) enable reactive bimanual control despite the computational cost of large VLA models.

**Generalization.** Hierarchical VLAs ( $\pi_{0.5}$  [3], **Hi Robot** [29]) demonstrate the strongest generalization to novel environments and open-ended instructions. Cross-embodiment pre-training on OXE data provides a foundation for transfer, though bimanual-specific skills require task-specific fine-tuning.

**Efficiency.** The computational cost of VLA inference remains a concern for bimanual real-time control. Table 15 summarizes the efficiency characteristics of representative methods. Flow-based models with  $K = 10$  denoising steps achieve the best latency-quality tradeoff, while efficient architectures (**TinyVLA** [7], **MiniVLA** [76]) sacrifice some capability for deployment on resource-constrained hardware.

**Memory.** Memory-augmented VLAs represent the most significant recent advance for long-horizon bimanual tasks. **MEM** [77], integrated into the  $\pi_{0.6}$  VLA, enables tasks spanning up to 15 minutes by combining video-based short-horizon memory with language-based long-horizon memory. Concurrent approaches [78,79,122] explore complementary designs, from perceptual-cognitive memory banks to amortized multi-frame context. The common finding is that different time scales demand different memory representations.

**World Models.** World model-powered VLAs [51,80] and direct video-action models [81,168] represent a shift from purely reactive policies to predictive ones. The GigaBrain family demonstrated that world model-generated data improves bimanual task performance by  $\sim 30\%$ , and web-scale video pre-training transfers to robot control with 10–20 hours of task-specific data. However, none of these approaches have been evaluated on standardized bimanual benchmarks, limiting direct comparison.

**Unmanned Aerial Robotics.** VLA adoption for drones lags manipulation by  $\sim 2$  years. Strong individual components exist (RL flight policies [137], VLN benchmarks [129], LLM-based planning [130]), but integrated end-to-end VLA systems for physical drones remain rare. The architectural innovations proven for bimanual manipulation are directly applicable (see directions 11–13 below).

**Industrial VLA Deployment.** The transition from research to product-level systems has accelerated in 2025–2026, with several companies deploying VLA-based robots commercially (Table 14). Three architectural patterns have emerged from industry that differ from the research approach.

First, *dual-system (S1/S2) architectures* separate slow reasoning from fast action. **Gemini Robotics** [178] runs a distilled VLM in the cloud ( $< 160$  ms query latency) paired with a local action decoder achieving 50 Hz control, and more than doubles performance on a broad generalization benchmark compared to other VLAs. **GR00T N1** [179] pairs a 1.34B-parameter VLM (System 2, 10 Hz) with a diffusion Transformer action head (System 1, 120 Hz). **Helix** [180] from Figure AI uses a 7B VLM at 7 Hz and an 80M-parameter action model at 200 Hz, controlling 35 degrees of freedom on embedded GPUs without cloud dependency. This pattern resolves the latency-capability tradeoff that limits monolithic VLAs (Direction 3).

Second, *video-as-action models* bypass direct action regression entirely. **Rhoda AI's** FutureVision [81] pre-trains a causal video model on web-scale video, predicts future frames conditioned on the current scene, and extracts actions via a learned inverse dynamics model. **1X Technologies** [181] takes a similar approach with a 14B-parameter text-conditioned diffusion world model trained on 900 hours of egocentric human video plus 70 hours of robot data. Both systems achieve one-shot or few-shot task adaptation from human demonstrations injected into the context window, without retraining.

Third, *continuous autonomous improvement* has reached production scale. **Dyna Robotics** [182] deploys a dual-arm foundation model (DYNA-1) with a proprietary reward model (informed by the Eureka [183] approach to LLM-generated rewards) that enables self-supervised error recovery. DYNA-1 achieves 99.4% success rate over 24+ hours of continuous autonomous operation in commercial settings, generating tens of terabytes of self-improvement data daily. This validates the RECAP paradigm (Section 6.3) at industrial scale. We note that industrial performance figures are self-reported under company-defined conditions and await independent replication.

Novel data collection strategies are also emerging. **Sunday Robotics** [184], founded by the creators of ALOHA [17] and Diffusion Policy [40], trains its ACT-1 foundation model on *zero robot data*. Instead, \$200 Skill Capture Gloves shipped to thousands of users collect 10 million human demonstration episodes across 500+ homes, with a learned Skill Transform layer adapting human kinematics to robot morphology. **Covariant's** RFM-1 [185], an 8B-parameter autoregressive world model, achieves 99%+ precision in warehouse picking by training on millions of real deployment interactions across hundreds of customer sites. **AgiBot World** [127] contributes 1M+ real robot trajectories across 217 tasks, with its GO-1 generalist policy outperforming Open X-Embodiment baselines by 30%.

Hardware-focused companies are also integrating VLA-class models. **Boston Dynamics** and Toyota Research Institute jointly developed a Large Behavior Model (LBM) [186] for the Atlas humanoid that controls the entire robot (hands and feet) through a single whole-body policy for packing, sorting, and organizing tasks. Atlas fleets are scheduled to ship to Hyundai and Google DeepMind in 2026, with Google's Gemini Robotics [178] being integrated for enhanced cognitive capabilities.

Open-source efforts from **NVIDIA** [179] (GR00T N1 weights and training data), **Xiaomi** [157] (Xiaomi-Robotics-0, LIBERO SOTA at 98.7%), and **AgiBot** [127] are accelerating community progress. Figure 11 shows representative systems from this industrial wave.



**Figure 11.** Industrial VLA-powered humanoid robot systems. (a, top left) Boston Dynamics Atlas with TRI Large Behavior Model performing warehouse manipulation [186]. (b, top right) Unitree humanoid executing dynamic whole-body control. (c, bottom left) Tesla Optimus humanoid with dexterous hands for general-purpose manipulation [187]. (d, bottom right) Google DeepMind Gemini Robotics, which integrates actions as a native Gemini modality for dexterous manipulation and general-purpose robot control [178]. Figures obtained from the respective organizations.

**Table 14.** Industrial VLA systems (2024–2026). Architecture indicates the action generation approach. Deployment column indicates the current operational status. Success rates are self-reported under company-defined conditions.

| System                                   | Organization          | Architecture                 | Key Innovation                          | Deployment        | Year |
|--|-----------------------|------------------------------|---|-------------------|------|
| <i>Dual-System (S1/S2) Architectures</i> |                       |                              |   |                   |      |
| Gemini Robotics [178]                    | Google DeepMind       | VLM + action decoder         | Actions as native Gemini modality       | Partner testing   | 2025 |
| GR00T N1 [179]                           | NVIDIA                | VLM (10 Hz) + DiT (120 Hz)   | Open-source; neural trajectory augment. | Research          | 2025 |
| Helix [180]                              | Figure AI             | VLM (7 Hz) + action (200 Hz) | 35-DOF on embedded GPU                  | BMW factory       | 2025 |
| <i>Video-as-Action / World Models</i>    |                       |                              |   |                   |      |
| FutureVision [81]                        | Rhoda AI              | Causal video → inv. dynamics | Web-scale video pre-training            | Industrial pilots | 2026 |
| 1XWM [181]                               | 1X Technologies       | Diffusion WM → IDM           | 900h human video + 70h robot            | Development       | 2025 |
| RFM-1 [185]                              | Covariant             | 8B AR world model            | 99%+ warehouse precision                | Hundreds of sites | 2024 |
| <i>Continuous Autonomous Improvement</i> |                       |                              |   |                   |      |
| DYNA-1 [182]                             | Dyna Robotics         | FM + proprietary RM          | 99.4% success, 24h autonomy             | Commercial sites  | 2025 |
| $\pi_0^*$ [4]                            | Physical Intelligence | FM + RECAP (RL)              | 10–40% over demo baseline               | Research          | 2025 |
| <i>Novel Data Collection</i>             |                       |                              |   |                   |      |
| ACT-1 [184]                              | Sunday Robotics       | Zero robot data; glove demos | 10M episodes from 500+ homes            | Beta 2026         | 2026 |
| AgiBot World [127]                       | AgiBot                | Latent action repr.          | 1M+ trajectories; 30% over OXE          | Shipping at scale | 2025 |
| <i>Open-Source VLAs</i>                  |                       |                              |   |                   |      |
| Xiaomi-Robotics-0 [157]                  | Xiaomi                | MoT + DiT (4.7B)             | LIBERO 98.7% SOTA                       | Open-source       | 2025 |
| GR00T N1 [179]                           | NVIDIA                | VLM + DiT (2.2B)             | Weights + data released                 | Open-source       | 2025 |

**Table 15.** Efficiency comparison of VLA models for bimanual deployment. GPU indicates the minimum GPU for real-time control.

| Method       | Params | Latency | Min. GPU |
|--------------|--------|---------|----------|
| RT-2 [1]     | 55B    | ~1 s    | TPU v4   |
| OpenVLA [5]  | 7B     | ~150 ms | A100     |
| $\pi_0$ [2]  | 3B     | ~70 ms  | A100     |
| RDT-1B [44]  | 1.2B   | ~150 ms | A6000    |
| TinyVLA [7]  | 1B     | ~40 ms  | RTX 4090 |
| MiniVLA [76] | 300M   | ~25 ms  | RTX 3090 |
| FAST [8]     | 7B     | ~80 ms  | A100     |

The field has also converged on several **long-standing challenges** that remain open despite recent progress:

- **Distribution shift:** VLA policies still degrade when encountering out-of-distribution observations, especially for bimanual tasks where object configurations have high variability.
- **Contact modeling:** Precise force control during bimanual contact is not addressed by current position-space VLAs.
- **Evaluation standardization:** The lack of common bimanual benchmarks prevents fair comparison across methods.
- **Data scarcity:** High-quality bimanual demonstrations remain expensive to collect, limiting the scale of bimanual VLA training.
- **Temporal credit assignment:** For long-horizon bimanual tasks, determining which actions contributed to success or failure is difficult, hindering RL-based improvement.

## 12.2. Summary and Discussion

In our view, the analysis yields the following key findings:

**(1) Flow matching is the current best action generation mechanism for bimanual VLAs.** We find that the combination of continuous action generation, efficient sampling ( $K = 10$  steps), and long action chunks makes flow matching uniquely suited to the high-dimensional, temporally correlated action spaces of bimanual manipulation. Diffusion models offer similar expressiveness but at higher computational cost.

**(2) VLM pre-training provides critical semantic grounding for bimanual tasks.** In our analysis, VLAs that inherit web-scale knowledge from VLM backbones consistently outperform architectures trained from scratch on robot data alone. The VLM’s understanding of objects, spatial relationships, and task semantics transfers directly to manipulation, reducing the amount of robot-specific data needed.

**(3) Action chunking is essential for bimanual coordination.** We observe that single-step action prediction cannot capture the coordinated motion patterns of two arms working in concert. Chunks of  $H = 50$  steps at 50 Hz (1 second of motion) provide sufficient temporal context for most bimanual primitives, including folding, handovers, and assembly.

**(4) Reinforcement learning from autonomous practice is, in our assessment, the single most impactful recent advance for bimanual VLAs.** RECAP [4] showed that VLAs can self-improve by practicing autonomously and learning from success/failure signals. This matters especially for bimanual tasks where demonstration data is expensive to collect and expert performance is difficult to achieve via teleoperation.

**(5) Hierarchical architectures enable long-horizon bimanual tasks.** We find that flat VLA policies struggle with tasks requiring more than a few steps of bimanual coordination. Hierarchical decomposition (high-level VLM reasoning plus low-level VLA execution) extends the effective planning horizon from seconds to minutes.

**(6) We find that data diversity matters more than data quantity for generalization.** VLAs pre-trained on diverse cross-embodiment data generalize better than those trained on larger quantities of homogeneous data. This finding has implications for bimanual data collection: collecting demonstrations across varied environments and objects is more valuable than maximizing episode count in a single setting.

**(7) The latency-reactivity tradeoff remains a fundamental challenge.** We note that large VLA models incur significant inference latency, conflicting with the need for reactive bimanual control. Techniques such as RTC [107], TinyVLA [7], and FAST [8] mitigate this but do not fully resolve it.

**(8) Bimanual benchmarks are insufficient.** We consider this the most pressing infrastructure gap in the field. Most VLA evaluation occurs on single-arm tasks or bespoke bimanual setups that vary across papers. Without standardized bimanual benchmarks, comparing methods fairly and tracking progress systematically is not possible.

**(9) Pre-training on web data transfers to bimanual tasks.** We observe that the semantic knowledge encoded in VLM backbones (object affordances, material properties, spatial reasoning) directly benefits bimanual manipulation, even though web data contains no robot actions. This transfer is most evident in language grounding (understanding what “fold” or “stack” means) and visual scene understanding (identifying object parts and configurations).

**(10) Bimanual coordination emerges from joint prediction.** We find this result surprising: VLAs that predict both arms’ actions jointly in a single action chunk learn coordination patterns implicitly from data, without explicit coordination mechanisms. This emergent coordination is strongest with flow-based and diffusion-based models that generate the full bimanual action in a single denoising process.

**(11) VLA architectures are cross-embodiment, with aerial applications lagging by ~2 years.** We observe that the same VLM backbones, action generation mechanisms, and training recipes that power bimanual manipulation are being adapted for unmanned aerial robotics. As of early 2026, the aerial VLA field is at the stage manipulation reached in 2022–2023: strong individual components exist but integrated end-to-end systems remain nascent. High-fidelity simulators, accessible hardware, and cross-embodiment pre-training provide the ingredients for rapid convergence.

**(12) Dual-system architectures are the industry consensus for product-level VLAs.** We find that Google (Gemini Robotics [178]), NVIDIA (GR00T N1 [179]), and Figure AI (Helix [180]) all independently converged on separating a slow reasoning module ( $\leq 10$  Hz) from a fast action module ( $\geq 100$  Hz). This pattern resolves the latency-capability tradeoff that monolithic VLAs face: the reasoning module provides semantic understanding and task decomposition, while the action module generates smooth, high-frequency motor commands. Research VLAs that adopt this pattern will be better positioned for deployment.

**(13) Production reliability requires continuous self-improvement, not just better demonstrations.** In our view, the most reliable deployed systems, Dyna’s DYNA-1 [182] (99.4% over 24 hours) and Covariant’s RFM-1 [185] (99%+ precision), achieve their performance through continuous RL loops where every deployment interaction feeds back into training, not through larger demonstration datasets alone. This validates the RECAP approach [4] at industrial scale and suggests that the path to product-level VLAs runs through autonomous improvement infrastructure.

**(14) Video prediction is gaining traction as an alternative to direct action regression.** We note that Rhoda AI’s FutureVision [81] and 1X Technologies’ world model [181] generate future video frames first and extract actions via inverse dynamics, exploiting web-scale video pre-training that contains orders of magnitude more data than robot demonstration datasets. This approach allows one-shot task adaptation from human demonstrations without retraining, though inference latency and physics fidelity remain open challenges.

### 12.3. Research Directions

Despite rapid progress, several fundamental challenges remain. We consider the following research directions most pressing:

**(1) Standardized Bimanual Benchmarks:** The field urgently needs standardized simulation and real-world benchmarks for bimanual manipulation, analogous to LIBERO for single-arm tasks. Such benchmarks should cover the full spectrum of coordination types (independent, loosely coupled, tightly coupled), object categories (rigid, articulated, deformable), and task horizons (single-step to multi-minute). Without standardized evaluation, comparing bimanual VLA methods remains unreliable.

**(2) Dexterous, Force-Aware, and Multi-Modal Manipulation:** Current VLA systems use parallel-jaw grippers and rely solely on visual observations, limiting both dexterity and contact awareness. Extending VLAs to multi-fingered hands would unlock tasks such as in-hand reorientation, but the action space (two 16-DOF hands plus two 7-DOF arms) exceeds 40 dimensions per step, posing extreme challenges for action generation. Simultaneously, incorporating force/torque feedback and tactile sensing (*e.g.*, GelSight) into VLA observations is essential for contact-rich tasks such as tightening screws, snapping parts, and kneading dough. Auditory signals can further complement vision for detecting task-relevant events such as clicks and snaps. Jointly addressing dexterity, force awareness, and multi-modal sensing is necessary to move bimanual VLAs beyond the current pick-and-place regime.

**(3) Real-Time Reactive Control:** Despite advances in RTC [107] and efficient architectures [7], attaining truly reactive bimanual control ( $> 100$  Hz) with large VLA models remains difficult. Research into model compression, speculative decoding for action generation, and hardware-software co-design could close this gap.

**(4) Data-Efficient Learning and Sim-to-Real Transfer:** Collecting bimanual demonstrations is expensive, and few-shot adaptation (fewer than 10 demonstrations) would reduce deployment costs. Cross-embodiment pre-training already provides strong priors; combining it with meta-learning, in-context learning, or skill composition could yield practical few-shot bimanual adaptation. Complementarily, simulation could provide unlimited training data, but the reality gap is severe for contact-rich bimanual tasks involving deformable objects. Advances in differentiable simulation, domain randomization tailored to bimanual contact, and sim-to-real fine-tuning are needed to unlock simulation as a primary data source.

**(5) Compositional Bimanual Skills:** Rather than learning each bimanual task from scratch, VLAs could learn a library of composable bimanual primitives (grasp, hold, fold, insert, handover) and combine them to perform novel tasks specified via language. Skill composition would improve generalization to unseen task combinations.

**(6) Safety-Certified Bimanual VLAs:** Deploying bimanual VLAs in human environments requires formal safety guarantees. Research into runtime monitoring, safety-constrained action generation, and provable collision avoidance between arms and with humans is essential. We believe this will become the primary bottleneck for commercial deployment, as current heuristic safety measures (rate limiting, action clipping) are insufficient for human-proximate operation.

**(7) Autonomous Improvement and World Models:** RECAP [4] demonstrated that VLAs can self-improve from autonomous practice, but the current approach requires human-designed task distributions and VLM-based reward signals that may not generalize. Integrating world models [51] that predict the consequences of bimanual actions, including object deformation and contact transitions, could enable look-ahead planning and more effective autonomous practice. The long-term goal is fully autonomous self-improvement where the VLA discovers new tasks, practices them, and improves without human oversight.

**(8) Human-Robot Collaborative Manipulation:** The ultimate bimanual system may involve one robot arm and one human arm working together. VLAs could learn to coordinate with a human partner, predicting human intentions and adapting robot actions accordingly. This requires advances in human motion prediction, shared autonomy, and real-time VLA adaptation.

**(9) Memory-Augmented VLAs for Long-Horizon Autonomy:** MEM [77] and concurrent work [78, 79, 119–123] show that multi-scale memory improves performance on tasks spanning minutes. Key

open challenges include scaling beyond single episodes to persistent deployment, learning what to remember versus forget, multi-modal memory grounding, and avoiding causal confusion. Persistent memory could enable continual learning across deployment sessions.

**(10) World Models for Bimanual Planning:** World models that predict future states [80,81,83,171,172] could enable look-ahead planning for contact-rich bimanual sequences. The GigaBrain family [51,80] improved bimanual task performance by  $\sim 30\%$  via world model-generated data, and video-action models [81,168,170] transfer web-scale video pre-training to robot control. Key challenges include predicting joint consequences of two coordinated arms on deformable objects and integrating predictions with real-time control. Unifying world models with memory (Direction 9) is promising: short-term prediction plus long-term state tracking.

**(11) End-to-End VLAs for Drone Control:** Building VLAs that map onboard camera images and language to continuous flight commands for physical drones is the most pressing aerial direction. Key challenges:  $\geq 100$  Hz latency requirements, outdoor 3D observation spaces, and the sim-to-real gap for underactuated dynamics. Efficient manipulation VLA architectures (TinyVLA, MiniVLA, FAST) are directly relevant given constrained onboard compute.

**(12) Multi-Agent Aerial VLAs:** Multi-drone coordination presents a natural extension of the bimanual coordination strategies analyzed in Section 8.1. Centralized VLAs that jointly generate actions for multiple drones face the same dimensional scaling challenges as bimanual joint action spaces, while decentralized approaches require explicit communication protocols. The hierarchical VLA paradigm (high-level VLM planner assigning subgoals to individual drone policies) is especially promising for heterogeneous multi-agent systems that combine aerial and ground robots.

**(13) Aerial Manipulation with VLAs:** Drones equipped with grippers or robotic arms must simultaneously stabilize flight and execute precise manipulation, combining the challenges of both domains surveyed in this paper. VLA architectures that generate coupled flight-and-grasp action chunks, analogous to bimanual joint action spaces, could enable aerial grasping, payload handover, and contact-based inspection tasks that are currently beyond the reach of separate flight and manipulation controllers.

**(14) Bridging the Research-to-Production Gap:** As Table 14 shows, industry has converged on architectural and training patterns that differ from the dominant research approach. Three gaps are most pressing. First, *sustained reliability*: research VLAs are evaluated over tens of trials, while production requires 99%+ success over thousands of continuous cycles; Dyna's DYNA-1 [182] and Covariant's RFM-1 [185] achieve this through continuous RL self-improvement loops that generate terabytes of training data daily. Second, *dual-system design*: the S1/S2 separation adopted by Gemini Robotics [178], GR00T N1 [179], and Helix [180] resolves the latency-capability tradeoff, but research on how to optimally partition reasoning and action across the two systems is nascent. Third, *scalable data collection*: Sunday Robotics' \$200 gloves [184] (10M episodes from 500+ homes), NVIDIA's neural trajectory augmentation [179] ( $10\times$  synthetic data expansion), and 1X's video-to-action pipeline [181] (900 hours of human video) each demonstrate that the demonstration bottleneck can be bypassed, but no unified framework exists. Research that addresses these three gaps (evaluation at production scale, principled S1/S2 co-design, and demonstration-free data scaling) will have the most direct path to real-world impact.

Table 16 maps each research direction to the VLA components and sections most relevant to its development.

VLA models have transformed bimanual manipulation in under three years, progressing from proof-of-concept demonstrations to autonomous household and industrial operation. The cross-embodiment nature of VLAs means that progress in manipulation accelerates unmanned aerial robotics and vice versa, while industry deployment is validating and reshaping research priorities in real time. The fourteen research directions above provide a roadmap for addressing the remaining gaps: standardized evaluation, dexterous force-aware control, memory and world models for long-horizon

planning, end-to-end drone VLAs, and bridging the widening gap between research benchmarks and production reliability.

**Table 16.** Summary of research directions with associated VLA components, current gap severity, and relevant review sections.

| #  | Direction                                  | Primary Component       | Gap Severity | Sections |
|----|--|-------------------------|--------------|----------|
| 1  | Standardized bimanual benchmarks           | Evaluation              | Critical     | 4, 8     |
| 2  | Dexterous, force-aware, multi-modal manip. | Observation/Action      | High         | 7, 8, 11 |
| 3  | Real-time reactive control                 | Architecture/Efficiency | Medium       | 5, 7     |
| 4  | Data-efficient learning & sim-to-real      | Training/Data           | High         | 6, 11    |
| 5  | Compositional bimanual skills              | Architecture/Language   | Medium       | 10, 8    |
| 6  | Safety-certified VLAs                      | Deployment              | Critical     | 11       |
| 7  | Autonomous improvement & world models      | Training/RL             | Medium       | 6, 5     |
| 8  | Human-robot collaboration                  | HRI                     | Low          | 11       |
| 9  | Memory-augmented long-horizon VLAs         | Architecture/Memory     | High         | 8, 5     |
| 10 | World models & future state prediction     | World Model/Planning    | High         | 11, 6    |
| 11 | End-to-end VLAs for drone control          | Architecture/Aerial     | Critical     | 9, 5     |
| 12 | Multi-agent aerial VLAs                    | Coordination/Aerial     | High         | 9, 8     |
| 13 | Aerial manipulation with VLAs              | Aerial/Manipulation     | High         | 9, 8     |
| 14 | Bridging research-to-production gap        | Deployment/Training     | Critical     | 12       |

**Author Contributions:** Conceptualization, I.S. and H.S.A.; methodology, I.S.; writing—original draft preparation, I.S.; writing—review and editing, H.S.A.; supervision, H.S.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable. This is a review article and no new data were created.

**Conflicts of Interest:** The authors declare no conflicts of interest.

**Acknowledgments:** The authors thank the open-source robotics and machine learning communities for making this rapidly evolving field accessible through shared code, models, and datasets. This work is additionally supported by Chef Robotics.

## Abbreviations

The following abbreviations are used in this manuscript:

|       |   |
|-------|---|
| VLA   | Vision-Language-Action model                      |
| VLM   | Vision-Language Model                             |
| BC    | Behavioral Cloning                                |
| IL    | Imitation Learning                                |
| RL    | Reinforcement Learning                            |
| FM    | Flow Matching                                     |
| ODE   | Ordinary Differential Equation                    |
| DOF   | Degrees of Freedom                                |
| OXE   | Open X-Embodiment                                 |
| AR    | Autoregressive                                    |
| DiT   | Diffusion Transformer                             |
| RECAP | Reinforcement Learning from Autonomous CAPability |

|      |                                   |
|------|-----------------------------------|
| RTC  | Real-Time Chunking                |
| TTAC | Training-Time Action Conditioning |
| BID  | Bidirectional Decoding            |
| DVA  | Direct Video Action               |
| WM   | World Model                       |
| MEM  | Multi-Scale Embodied Memory       |
| UAV  | Unmanned Aerial Vehicle           |
| UGV  | Unmanned Ground Vehicle           |
| VLN  | Vision-Language Navigation        |
| MAV  | Micro Aerial Vehicle              |
| IMU  | Inertial Measurement Unit         |

## References

1. Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Chormanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. *arXiv preprint arXiv:2307.15818* **2023**.
2. Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al.  $\pi_0$ : A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164* **2024**.
3. Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al.  $\pi_{0.5}$ : A Vision-Language-Action Model with Open-World Generalization. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2025.
4. Amin, R.; Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; et al.  $\pi_{0.6}^*$ : A VLA That Learns From Experience. *arXiv preprint arXiv:2511.14759* **2025**.
5. Kim, M.J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Nasiriany, M.; et al. OpenVLA: An Open-Source Vision-Language-Action Model. *arXiv preprint arXiv:2406.09246* **2024**.
6. Octo Model Team.; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Kreiman, T.; Xu, C.; et al. Octo: An Open-Source Generalist Robot Policy. In Proceedings of the Proceedings of Robotics: Science and Systems (RSS), 2024.
7. Wen, J.; Zhu, Y.; Zhang, J.; Mu, M.; Qi, Z.; Peng, Z.; Wan, G.; Li, T.; Huang, J.; Lu, H. TinyVLA: Towards Fast, Data-Efficient Vision-Language-Action Models for Robotic Manipulation. *arXiv preprint arXiv:2409.12514* **2024**.
8. Pertsch, K.; Kim, M.J.; Luo, J.; Levine, S.; Finn, C. FAST: Efficient Action Tokenization for Vision-Language-Action Models. In Proceedings of the Proceedings of Robotics: Science and Systems (RSS), 2025.
9. Arshad, A.; Jia, X.; Sun, L. CognitiveDrone: A VLA Model and Evaluation Benchmark for Real-Time Cognitive Task Solving and Reasoning in UAVs. *arXiv preprint arXiv:2503.01378* **2025**.
10. Saha, A.; Mishra, S.; Fang, Y.; Liu, M.; Xu, D.; Garg, A. DroneVLA: Vision-Language-Action Model for Aerial Manipulation. *arXiv preprint arXiv:2601.13809* **2026**.
11. Liu, C.; Chen, Z.; Zhao, Y.; Liu, M.; Luo, W. AIR-VLA: End-to-End Vision-Language-Action Model for Aerial Manipulation. *arXiv preprint arXiv:2601.21602* **2026**.
12. Zhou, G.; Li, Y.C.; Pan, Y.; Huang, Z.; Lin, Y.A.; Xu, J.; Song, S. Flying Hand: End-Effector-Centric Framework for Versatile Aerial Manipulation Imitation Learning. *arXiv preprint arXiv:2504.10334* **2025**.
13. Firoozi, R.; Tucker, J.; Tian, S.; Majumdar, A.; Sun, J.; Liu, W.; Zhu, Y.; Song, S.; Kapoor, A.; Hausman, K.; et al. Foundation Models in Robotics: Applications, Challenges, and the Future. *The International Journal of Robotics Research* **2024**, *43*, 2164–2204.
14. Wolf, R.; Shi, Y.; Liu, S.; Rayyes, R. Diffusion Models for Robotic Manipulation: A Survey. *Frontiers in Robotics and AI* **2025**, *12*.
15. Abbas, M.; Narayan, J.; Dwivedy, S.K. A Systematic Review on Cooperative Dual-Arm Manipulators: Modeling, Planning, Control, and Vision Strategies. *International Journal of Intelligent Robotics and Applications* **2023**, *7*, 683–707.
16. Kreiman, T.; Levine, S.; Finn, C. Bimanual Coordination for Robot Manipulation: A Survey. *arXiv preprint* **2024**.
17. Zhao, T.Z.; Kumar, V.; Levine, S.; Finn, C. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. *arXiv preprint arXiv:2304.13705* **2023**.

18. Lipman, Y.; Chen, R.T.Q.; Ben-Hamu, H.; Nickel, M. Flow Matching for Generative Modeling. *arXiv preprint arXiv:2210.02747* **2022**.
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2017.
20. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the Proceedings of the International Conference on Learning Representations (ICLR), 2021.
21. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the Proceedings of the International Conference on Machine Learning (ICML), 2021.
22. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2020.
23. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training Language Models to Follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems (NeurIPS)* **2022**.
24. Driess, D.; Xia, F.; Sajjadi, M.S.M.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; et al. PaLM-E: An Embodied Multimodal Language Model. In Proceedings of the Proceedings of the International Conference on Machine Learning (ICML), 2023.
25. Beyer, L.; Steiner, A.; Pinto, A.S.; Kolesnikov, A.; Wang, X.; Salz, D.; Neumann, M.; Alabdulmohsin, I.; Tschannen, M.; Bugliarello, E.; et al. PaLI Gemma: A Versatile 3B VLM for Transfer. *arXiv preprint arXiv:2407.07726* **2024**.
26. Gemma Team. Gemma: Open Models Based on Gemini Research and Technology. *arXiv preprint arXiv:2403.08295* **2024**.
27. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971* **2023**.
28. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual Instruction Tuning. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2024.
29. Shi, L.X.; Ichter, B.; Equi, M.; Levine, S.; Hausman, K. Hi Robot: Open-Ended Instruction Following with Hierarchical Vision-Language-Action Models. In Proceedings of the Proceedings of the International Conference on Machine Learning (ICML), 2025.
30. Pomerleau, D.A. ALVINN: An Autonomous Land Vehicle in a Neural Network. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 1989.
31. Ross, S.; Gordon, G.J.; Bagnell, D. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In Proceedings of the Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), 2011.
32. Stepputtis, S.; Campbell, J.; Phielipp, M.; Lee, S.; Baral, C.; Ben Amor, H. Language-Conditioned Imitation Learning for Robot Manipulation Tasks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2020.
33. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the Proceedings of the International Conference on Learning Representations (ICLR), 2014.
34. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2014.
35. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2020.
36. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In Proceedings of the Proceedings of the International Conference on Learning Representations (ICLR), 2021.
37. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

38. Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; Mordatch, I. Decision Transformer: Reinforcement Learning via Sequence Modeling. *Advances in Neural Information Processing Systems (NeurIPS)* **2021**.
39. Reed, S.; Zolna, K.; Parisotto, E.; Colmenarejo, S.G.; Novikov, A.; Barth-Maron, G.; Giménez, M.; Sulsky, Y.; Kay, J.; Springenberg, J.T.; et al. A Generalist Agent. In Proceedings of the Transactions on Machine Learning Research (TMLR), 2022.
40. Chi, C.; Feng, S.; Du, Y.; Xu, Z.; Cousineau, E.; Burchfiel, B.; Song, S. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *The International Journal of Robotics Research (IJRR)* **2023**.
41. Liu, X.; Gong, C.; Liu, Q. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. *arXiv preprint arXiv:2209.14577* **2022**.
42. Fu, Z.; Zhao, T.Z.; Finn, C. Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2024.
43. Chi, C.; Xu, Z.; Pan, C.; Cousineau, E.; Burchfiel, B.; Feng, S.; Tedrake, R.; Song, S. Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots. In Proceedings of the Proceedings of Robotics: Science and Systems (RSS), 2024.
44. Liu, S.; Wu, L.; Li, B.; Tan, H.; Chen, H.; Wang, Z.; Xu, K.; Su, H.; Zhu, J. RDT-1B: A Diffusion Foundation Model for Bimanual Manipulation. *arXiv preprint arXiv:2410.07864* **2024**.
45. Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In Proceedings of the Field and Service Robotics (FSR), 2018.
46. Song, Y.; Naji, S.; Kaufmann, E.; Loquercio, A.; Scaramuzza, D. Flightmare: A Flexible Quadrotor Simulator. *Conference on Robot Learning (CoRL)* **2021**.
47. Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. *arXiv preprint arXiv:2310.08864* **2024**.
48. Khazatsky, A.; Pertsch, K.; Nair, S.; Balakrishna, A.; Dasari, S.; Karamcheti, S.; Nasiriany, M.; Srirama, M.K.; Chen, L.Y.; Ellis, K.; et al. DROID: A Large-Scale In-the-Wild Robot Manipulation Dataset. *arXiv preprint arXiv:2403.12945* **2024**.
49. Walke, H.; Black, K.; Zhao, T.Z.; Vuong, Q.; Zheng, C.; Florence, P.; Levine, S. BridgeData V2: A Dataset for Robot Learning at Scale. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2023.
50. Ebert, F.; Yang, Y.; Schmeckpeper, K.; Buber, B.; Georgakis, G.; Daniilidis, K.; Finn, C.; Levine, S. Bridge Data: Boosting Generalization of Robotic Skills with Cross-Domain Datasets. In Proceedings of the Proceedings of Robotics: Science and Systems (RSS), 2021.
51. GigaAI. GigaBrain-0.5M\*: A VLA with World Model-Based Reinforcement Learning. *arXiv preprint arXiv:2602.12099* **2025**.
52. Liu, B.; Zhu, Y.; Gao, C.; Feng, Y.; Liu, Q.; Zhu, Y.; Stone, P. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. *Advances in Neural Information Processing Systems (NeurIPS)* **2024**.
53. Li, X.; Hsu, K.; Liu, J.; Pertsch, K.; Vuong, Q.; Levine, S. SIMPLER: Simulated Manipulation Policy Evaluation for Real Robot Setups. *arXiv preprint* **2024**.
54. James, S.; Ma, Z.; Arrojo, D.R.; Davison, A.J. RL Bench: The Robot Learning Benchmark and Learning Environment. In Proceedings of the IEEE Robotics and Automation Letters (RA-L), 2020.
55. Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; Levine, S. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. *Proceedings of the Conference on Robot Learning (CoRL)* **2020**.
56. Zhu, Y.; Wong, J.; Mandlekar, A.; Martín-Martín, R.; Joshi, A.; Nasiriany, S.; Zhu, Y. robosuite: A Modular Simulation Framework and Benchmark for Robot Learning. In Proceedings of the arXiv preprint arXiv:2009.12293, 2020.
57. Gu, J.; Xiang, F.; Li, X.; Ling, Z.; Liu, X.; Mu, T.; Tang, Y.; Tao, S.; Wei, X.; Yao, Y.; et al. ManiSkill2: A Unified Benchmark for Generalizable Manipulation Skills. In Proceedings of the Proceedings of the International Conference on Learning Representations (ICLR), 2023.
58. Lee, C.; Zhang, R.; Zhu, J.; Xia, F.; Ehsani, K.; Martín-Martín, R.; Li, Y.; Savarese, S.; Fei-Fei, L. BEHAVIOR-1K: A Human-Centered, Embodied AI Benchmark with 1,000 Everyday Activities and Realistic Simulation. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2024.
59. Li, X.; Hsu, K.; Liu, J.; Pertsch, K.; Vuong, Q.; Levine, S. Evaluating Real-World Robot Manipulation Policies in Simulation. *arXiv preprint* **2024**.

60. Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Guber, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. RT-1: Robotics Transformer for Real-World Control at Scale. *arXiv preprint arXiv:2212.06817* **2022**.
61. Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohlhart, P.; Welker, S.; Wahid, A.; et al. RT-2-X: Learning Robot Skills from Large-Scale Data. In Proceedings of the arXiv preprint, 2023.
62. Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In Proceedings of the Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2024.
63. Zheng, J.; Kim, M.J.; Pertsch, K.; Karamcheti, S.; Finn, C.; Levine, S.; Liang, P. OpenVLA 2.0: Advancing Vision-Language-Action Models with Updated Training Recipes and Data. *arXiv preprint* **2024**.
64. Wu, K.; Wu, X.; Zhang, R.; Duan, J.; Ni, B.; Lu, J.; Zheng, J.; Fan, H. GR-1: Unleashing Large-Scale Video Generative Pre-Training for Visual Robot Manipulation. In Proceedings of the arXiv preprint, 2023.
65. Li, Y.; Fang, K.; Hausman, K.; Ichter, B.; Florence, P. Hamster: Hierarchical Action Models for Open-World Robot Manipulation. *arXiv preprint* **2025**.
66. Chen, D.; Wang, J.; Xu, Y.; Zhang, R.; Li, X.; Wu, Y.; Shao, J.; Ke, L. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model. *arXiv preprint* **2024**.
67. Haldar, S.; Pari, J.; Rao, A.; Watter, M.; Pinto, L. BAKU: An Efficient Transformer for Multi-Task Policy Learning. In Proceedings of the arXiv preprint, 2024.
68. Kim, M.J.; Pertsch, K.; Sadigh, D.; Levine, S.; Finn, C. KAT: Keypoint-Action Tokens for Robot Manipulation. *arXiv preprint* **2024**.
69. Zhen, H.; Qiu, Y.; Sun, S. SimpleVLA-RL: Simple Vision-Language-Action Models Meet Reinforcement Learning. *arXiv preprint* **2024**.
70. Li, Q.; Liang, Y.; Wang, Z.; Luo, L.; Chen, X.; Fang, M.; Jiang, J.; Fan, H.; Duan, N. CogACT: A Foundational Vision-Language-Action Model for Synergizing Cognition and Action in Robotic Manipulation. *arXiv preprint* **2025**.
71. Shridhar, M.; Manuelli, L.; Fox, D. Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2023.
72. Goyal, A.; Xu, J.; Guo, Y.; Blukis, V.; Chao, Y.W.; Fox, D. RVT: Robotic View Transformer for 3D Object Manipulation. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2023.
73. Ze, Y.; Yan, G.; Wu, Y.; Jia, Y.; Zhang, R.; Hu, Y.; Wu, J.; Tan, J.; Sun, H.; Su, H.; et al. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. In Proceedings of the Proceedings of Robotics: Science and Systems (RSS), 2024.
74. Zhou, C.; Yu, L.; Babu, A.; Tirumala, K.; Yasunaga, M.; Shamber, L.; Kahn, J.; Ma, X.; Zettlemoyer, L.; Levy, O. Transfusion: Predict the Next Token and Diffuse Images with One Multi-Modal Model. *arXiv preprint arXiv:2408.11039* **2024**.
75. Chen, J.; Li, S.; Zhang, K.; Chen, Y.; Ding, R.; Ge, Y.; Zhao, J. HybridVLA: Collaborative Diffusion and Autoregression in a Unified Vision-Language-Action Model. *arXiv preprint arXiv:2503.10631* **2025**.
76. Belkhale, S.; Sadigh, D. MiniVLA: A Better VLA with a Smaller Footprint. *arXiv preprint* **2024**.
77. Torne, M.; Pertsch, K.; Walke, H.; Vedder, K.; Nair, S.; Ichter, B.; Ren, A.Z.; Wang, H.; Tang, J.; Stachowicz, K.; et al. MEM: Multi-Scale Embodied Memory for Vision Language Action Models. *arXiv preprint arXiv:2503.02760* **2026**.
78. Shi, H.; Bin, X.; Liu, Y.; Sun, L.; Fengrong, L.; Wang, T.; Zhou, E.; Fan, H.; Zhang, X.; Huang, G. MemoryVLA: Perceptual-Cognitive Memory in Vision-Language-Action Models for Robotic Manipulation. *arXiv preprint arXiv:2508.19236* **2025**.
79. Jang, H.; Yu, S.; Kwon, H.; Jeon, H.; Seo, Y.; Shin, J. ContextVLA: Vision-Language-Action Model with Amortized Multi-Frame Context. *arXiv preprint arXiv:2510.04246* **2025**.
80. GigaAI Team. GigaBrain-0: A World Model-Powered Vision-Language-Action Model. *arXiv preprint arXiv:2510.19430* **2025**.
81. Rhoda AI. Causal Video Models Are Data-Efficient Robot Policy Learners. <https://www.rhoda.ai/research/direct-video-action>, 2026.
82. WorldVLA Team. WorldVLA: Towards Autoregressive Action World Model. *arXiv preprint arXiv:2506.21539* **2025**.
83. GigaAI Team. GigaWorld-0: World Models as Data Engine to Empower Embodied AI. *arXiv preprint arXiv:2511.19861* **2025**.
84. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *Proceedings of the International Conference on Learning Representations (ICLR)* **2022**.

85. Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C.D.; Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *Advances in Neural Information Processing Systems (NeurIPS)* **2023**.
86. Driess, D.; Finn, C.; Levine, S. Knowledge Insulation for Task-Oriented Fine-Tuning of Vision-Language-Action Models. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2025.
87. Xiao, M.; Song, Z.; Sebe, N.; Cheng, L. Align-then-Steer: Aligning Vision-Language-Action Models for Efficient Robot Policy Fine-Tuning. *arXiv preprint* **2025**.
88. Mandlkar, A.; Xu, D.; Wong, J.; Nasiriany, S.; Wang, C.; Kulkarni, R.; Fei-Fei, L.; Savarese, S.; Zhu, Y.; Martín-Martín, R. What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2022.
89. Bharadhwaj, H.; Vakil, J.; Sharma, M.; Gupta, A.; Tulsiani, S.; Kumar, V. RoboAgent: Generalization and Efficiency in Robot Manipulation via Semantic Augmentations and Action Chunking. In Proceedings of the Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2024.
90. Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; et al. QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2018.
91. Kumar, A.; Zhou, A.; Tucker, G.; Levine, S. Conservative Q-Learning for Offline Reinforcement Learning. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2020.
92. Peng, X.B.; Kumar, A.; Zhang, G.; Levine, S. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. In Proceedings of the arXiv preprint arXiv:1910.00177, 2019.
93. Levine, S.; Kumar, A.; Tucker, G.; Fu, J. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv preprint arXiv:2005.01643* **2020**.
94. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. In Proceedings of the arXiv preprint arXiv:1707.06347, 2017.
95. Ranawaka Aracchige, R.; Chi, C.; Song, S.; Burchfiel, B. SAIL: Sample-Efficient Policy Adaptation for Faster-than-Demonstration Execution. *arXiv preprint* **2025**.
96. Lu, J.; Luo, J.; Pertsch, K.; Levine, S. VLA-RL: Reinforcement Learning for Vision-Language-Action Models. *arXiv preprint* **2025**.
97. Chen, J.; Yuan, Y.; Li, J.; Qiao, Y. ConRFT: A Reinforced Fine-Tuning Method for VLA Models via Consistency Regularization. *arXiv preprint* **2025**.
98. Chebotar, Y.; Vuong, Q.; Hausman, K.; Xia, F.; Lu, Y.; Irpan, A.; Kumar, A.; Yu, T.; Herzog, A.; Pertsch, K.; et al. Q-Transformer: Scalable Offline Reinforcement Learning via Autoregressive Q-Functions. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2023.
99. Ren, A.Z.; Dinh, J.; Dai, S.; Zhang, T.; Phielipp, M.; Burchfiel, B. Diffusion Policy Policy Optimization. *arXiv preprint* **2023**.
100. Ghasemipour, S.K.S.; Florence, P.; Lazzaro, S.; Levine, S. Self-Improving Foundation Models for Embodied Intelligence. *arXiv preprint* **2025**.
101. Levine, S.; Pastor, P.; Krizhevsky, A.; Ibarz, J.; Quillen, D. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection. *The International Journal of Robotics Research (IJRR)* **2018**, *37*, 421–436.
102. Ha, H.; Florence, P.; Song, S. Scaling Up and Distilling Down: Language-Guided Robot Skill Acquisition. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2023.
103. Zheng, J.; Yao, J.; Yu, D.; Zhao, Z.; Wang, T.; Pan, L.; Zhang, L.; Yang, Y.; Zou, J. Universal Actions for Enhanced Embodied Foundation Models. *arXiv preprint* **2025**.
104. Ke, L.; Pertsch, K.; Levine, S.; Finn, C. Consistency Models as a Rich and Efficient Policy Class for Reinforcement Learning. In Proceedings of the arXiv preprint, 2024.
105. Zhao, T.Z.; Kumar, V.; Levine, S.; Finn, C. Learning Bimanual Manipulation with Action Chunking Transformers. In Proceedings of the Proceedings of Robotics: Science and Systems (RSS), 2024.
106. Dai, Y.; Bahl, S.; Singh, A.; Pertsch, K.; Levine, S. RACER: Rich Language-Guided Failure Recovery Policies for Imitation Learning. *arXiv preprint* **2024**.
107. Black, K.; Pertsch, K.; Nair, S.; Levine, S. Real-Time Chunking: Real-Time Execution of Action Chunking Flow Policies. *arXiv preprint* **2025**.
108. Liu, Y.; Hamdi, A.; Gkanatsios, N.; Fragkiadaki, K. Bidirectional Decoding: Improving Action Chunking via Closed-Loop Resampling. *arXiv preprint* **2024**.

109. Black, K.; Pertsch, K.; Nair, S.; Levine, S. Training-Time Action Conditioning for Efficient Real-Time Chunking. *arXiv preprint* **2025**.
110. Grannen, J.; Chong, Y.; Zhao, T.Z.; Finn, C. Stabilize to Act: Learning to Coordinate for Bimanual Manipulation. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2023.
111. Chitnis, R.; Tulsiani, S.; Gupta, S.; Gupta, A. Efficient Bimanual Manipulation Using Learned Task Schemas. In Proceedings of the Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2020.
112. Liang, J.; Huang, W.; Xia, F.; Xu, P.; Hausman, K.; Ichter, B.; Florence, P.; Zeng, A. Code as Policies: Language Model Programs for Embodied Control. In Proceedings of the Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2023.
113. Ahn, M.; Brohan, A.; Brown, N.; Chebotar, Y.; Cortes, O.; David, B.; Finn, C.; Fu, C.; Goyer, K.; Hausman, K.; et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2022.
114. Wang, C.; Fan, L.; Sun, J.; Zhang, R.; Fei-Fei, L.; Xu, D.; Zhu, Y.; Anandkumar, A. MimicPlay: Long-Horizon Imitation Learning by Watching Human Play. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2023.
115. Chen, L.; Bahl, S.; Pathak, D. PlayFusion: Skill Acquisition via Diffusion from Language-Annotated Play. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2023.
116. Du, Y.; Yang, S.; Dai, B.; Dai, H.; Nachum, O.; Tompson, J.; Schuurmans, D.; Abbeel, P. Learning Universal Policies via Text-Guided Video Generation. *Advances in Neural Information Processing Systems (NeurIPS)* **2024**.
117. Hu, Y.; Xie, F.; Jia, W.; Wang, G.; Zhao, J.; Gao, Y. Look Before You Leap: Unveiling the Power of GPT-4V in Robotic Vision-Language Planning. In Proceedings of the arXiv preprint, 2023.
118. Ma, Y.J.; Liang, W.; Wang, G.; Huang, D.A.; Bastani, O.; Jayaraman, D.; Zhu, Y.; Fan, L.; Anandkumar, A. Leveraging Internet-Scale Robotic Manipulation Data for Robotic Grasping. In Proceedings of the arXiv preprint, 2024.
119. Li, H.S.; Yang, Y.; Chen, X.; Chen, X.; Yang, Y.; Tian, H.; Wang, T.; Lin, D.; Zhao, F. CronusVLA: Towards Efficient and Robust Manipulation via Multi-Frame Vision-Language-Action Modeling. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025.
120. Mark, M.S.; Liang, J.; Attarian, M.; Fu, C.; Dwibedi, D.; Shah, D.; Kumar, A. BPP: Long-Context Robot Imitation Learning by Focusing on Key History Frames. *arXiv preprint arXiv:2602.15010* **2026**.
121. Torne, M.; Tang, A.; Liu, Y.; Finn, C. Learning Long-Context Diffusion Policies via Past-Token Prediction. *arXiv preprint arXiv:2505.09561* **2025**.
122. Fang, H.; Grotz, M.; Pumacay, W.; Wang, Y.R.; Fox, D.; Krishna, R.; Duan, J. SAM2Act: Integrating Visual Foundation Model with a Memory Architecture for Robotic Manipulation. In Proceedings of the Proceedings of the International Conference on Machine Learning (ICML), 2025.
123. Sridhar, A.; Pan, J.; Sharma, S.; Finn, C. MemER: Scaling Up Memory for Robot Control via Experience Retrieval. *arXiv preprint arXiv:2510.20328* **2025**.
124. Wei, Y.L.; Liao, H.; Lin, Y.; Wang, P.; Liang, Z.; Liu, G.; Zheng, W.S. CycleManip: Enabling Cyclic Task Manipulation via Effective Historical Perception and Understanding. *arXiv preprint arXiv:2512.01022* **2025**.
125. Chi, C.; Xu, Z.; Song, S. UMI on Legs: Making Manipulation Policies Mobile with Locomotion. *arXiv preprint* **2025**.
126. Shah, R.; Martín-Martín, R.; Zhu, Y. BUMBLE: Unifying Reasoning and Acting with Vision-Language Models for Building-wide Mobile Manipulation. *arXiv preprint* **2024**.
127. AgiBot Team. AgiBot World: A New Frontier for Generalist Robot Policies. *arXiv preprint* **2025**.
128. Physical Intelligence. Scaling Robot Policy Learning via Zero-Shot Labeling with Foundation Models. *arXiv preprint* **2025**.
129. Liu, S.; Zhang, H.; Li, Y. AerialVLN: Vision-and-Language Navigation for UAVs. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023.
130. Shah, D.; Equi, M.; Osinski, B.; Levine, S. Navigation with Large Language Models: Semantic Guessing as a Heuristic for Planning. In Proceedings of the Conference on Robot Learning (CoRL), 2023.
131. Zhang, W.; et al. SkyGPT: Autonomous UAV Navigation with Vision-Language Foundation Models. *arXiv preprint* **2025**.
132. Khasianov, A.; Manghi, T.; Nesterov, A.; Belousov, B.; Peters, J. UAV-VLA: Vision-Language-Action System for Large Scale Aerial Mission Generation. In Proceedings of the Companion of the ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2025. arXiv:2501.05014.

133. Gao, Z.; Wang, Y.; Sun, P. End-to-End Vision-Language Navigation for UAVs. *arXiv preprint arXiv:2504.21432* **2025**.
134. Gao, Y.; Wang, C.; Chen, Z.; Wang, Y.; Zhao, Y. OpenFly: A Versatile Toolchain and Large-Scale Benchmark for Aerial Vision-Language Navigation. *arXiv preprint* **2025**.
135. He, J.; Li, X.; Zhang, Y. CityNavAgent: Aerial Vision-Language Navigation with Hierarchical Semantic Planning and Global Memory. *arXiv preprint* **2025**.
136. Hwangbo, J.; Sa, I.; Siegwart, R.; Hutter, M. Control of a Quadrotor with Reinforcement Learning. *IEEE Robotics and Automation Letters* **2017**, *2*, 2096–2103.
137. Kaufmann, E.; Loquercio, A.; Ranftl, R.; Müller, M.; Koltun, V.; Scaramuzza, D. Champion-level drone racing using deep reinforcement learning. *Nature* **2023**, *620*, 982–987.
138. Shi, G.; Shi, X.; O’Connell, M.; Yu, R.; Azizzadenesheli, K.; Anandkumar, A.; Yue, Y.; Chung, S.J. Neural Lander: Stable Drone Landing Control Using Learned Dynamics. *IEEE International Conference on Robotics and Automation (ICRA)* **2019**.
139. Kim, H.; Park, J.; Scaramuzza, D. RaceVLA: VLA-Based Racing Drone with Human-Like Behavior. *arXiv preprint* **2025**.
140. Loquercio, A.; Kaufmann, E.; Scaramuzza, D. Dream to Fly: Model-Based Reinforcement Learning for Vision-Based Drone Flight. *arXiv preprint* **2025**.
141. O’Connell, M.; Shi, G.; Shi, X.; Azizzadenesheli, K.; Anandkumar, A.; Yue, Y.; Chung, S.J. Neural-Fly Enables Rapid Learning for Agile Flight in Strong Winds. *Science Robotics* **2022**, *7*.
142. Zhang, T.; et al. Learning to Fly by Grasping: Aerial Manipulation with Reinforcement Learning. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019.
143. Martinez, P.; Suarez, A.; Ollero, A. Avocado Harvesting with Aerial Bimanual Manipulation. *arXiv preprint arXiv:2408.09058* **2024**.
144. Zhang, Y.; et al. AeroAgent: Autonomous LLM-Based Drone Agent for Complex Mission Planning. *arXiv preprint* **2025**.
145. Chen, G.; Yao, X.; Yang, X.; Hu, Y.; Xu, J.; Zhang, Z. TypeFly: Flying Drones with Large Language Model. In Proceedings of the Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys), 2024.
146. Lynch, C.; Wahid, A.; Tompson, J.; Ding, T.; Betker, J.; Baruch, R.; Armstrong, T.; Florence, P. Interactive Language: Talking to Robots in Real Time. In Proceedings of the IEEE Robotics and Automation Letters (RA-L), 2023.
147. Zhang, K.; Yang, Z.; Başar, T. Decentralized Multi-Agent Reinforcement Learning for Multi-Robot Coordination. *Autonomous Robots* **2021**.
148. Ghosh, R.; Luo, J.; Geng, C.; Duan, Y.; Sadigh, D.; Finn, C.; Levine, S. Scaling Cross-Embodied Learning: One Policy for Manipulation, Navigation, Locomotion and Aviation. *arXiv preprint* **2024**.
149. Furrer, F.; Burri, M.; Achtelik, M.; Siegwart, R. RotorS: A Modular Gazebo MAV Simulator Framework. In Proceedings of the Robot Operating System (ROS): The Complete Reference, 2016.
150. Wang, W.; Zhu, D.; Wang, X.; Hu, Y.; Qiu, Y.; Wang, C.; Hu, Y.; Kapoor, A.; Scherer, S. TartanAir: A Dataset to Push the Limits of Visual SLAM. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* **2020**.
151. Fonder, M.; Van Droogenbroeck, M. Mid-Air: A Multi-Modal Dataset for Extremely Low Altitude Drone Flights. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* **2019**.
152. Jang, E.; Irpan, A.; Khansari, M.; Kappler, D.; Ebert, F.; Lynch, C.; Levine, S.; Finn, C. BC-Z: Zero-Shot Task Generalization with Robotic Imitation Learning. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2022.
153. Huang, W.; Wang, C.; Zhang, R.; Li, Y.; Wu, J.; Fei-Fei, L. VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models. *Proceedings of the Conference on Robot Learning (CoRL)* **2023**.
154. Mandlekar, A.; Nasiriany, S.; Wen, B.; Akinola, I.; Narang, Y.; Fan, L.; Zhu, Y.; Fox, D. Manipulate-Anything: Automating Real-World Robots using Vision-Language Models. *arXiv preprint* **2024**.
155. Duan, J.; Nasiriany, S.; Li, H.; Mandlekar, A. Manipulate-Anything: Automating Real-World Robots using Vision-Language Models. *arXiv preprint* **2024**.
156. Xian, Z.; Gkanatsios, N.; Gerber, T.; Ke, T.W.; Fragkiadaki, K. Chain-of-Thought Predictive Control. In Proceedings of the arXiv preprint, 2023.
157. Xiaomi Robotics Team. Xiaomi-Robotics-0: An Open-Source Vision-Language-Action Model with Real-Time Execution. *arXiv preprint* **2025**.

158. Brohan, A.; Chebotar, Y.; Finn, C.; Hausman, K.; Herzog, A.; Ho, D.; Ibarz, J.; Irpan, A.; Jang, E.; Julian, R.; et al. RT-H: Action Hierarchies Using Language. In Proceedings of the arXiv preprint, 2024.
159. Liu, P.; Orru, Y.; Paxton, C.; Shafiullah, N.M.M.; Pinto, L. OK-Robot: What Really Matters in Integrating Open-Knowledge Models for Robotics. *arXiv preprint* **2024**.
160. Etukuru, H.; Nair, S.; Pari, J.; Padmanabha, A.; Kamat, G.; Dasari, S.; Arbuckle, T.; Maddukuri, B.; Juliani, A.; Levine, S. Robot Utility Models: General Policies for Zero-Shot Deployment in New Environments. *arXiv preprint* **2024**.
161. Zeng, A.; Florence, P.; Tompson, J.; Welker, S.; Chien, J.; Attarian, M.; Armstrong, T.; Krasin, I.; Duong, D.; Sindhvani, V.; et al. Transporter Networks: Rearranging the Visual World for Robotic Manipulation. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2021.
162. Nair, S.; Rajeswaran, A.; Kumar, V.; Finn, C.; Gupta, A. R3M: A Universal Visual Representation for Robot Manipulation. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2022.
163. Nair, S.; Mitchell, E.; Chen, K.; Savarese, S.; Finn, C.; Sadigh, D. Learning Language-Conditioned Robot Behavior from Offline Data and Crowd-Sourced Annotation. In Proceedings of the Proceedings of the Conference on Robot Learning (CoRL), 2022.
164. Majumdar, A.; Yadav, K.; Arnaud, S.; Ma, Y.J.; Chen, C.; Silwal, S.; Jain, A.; Berber, V.P.; Mathur, P.; Olkin, T.; et al. Where Are We in the Search for an Artificial Visual Cortex for Embodied Intelligence? In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2023.
165. Cheng, H.; Cheang, C.; Li, Y.; Yang, J.; Lu, H. SPA: 3D Spatial-Awareness Enables Effective Embodied Representation. *arXiv preprint* **2024**.
166. Bharadhwaj, H.; Mottaghi, R.; Tulsiani, S.; Gupta, A. Gen2Act: Human Video Generation in Novel Scenarios Enables Generalizable Robot Manipulation. In Proceedings of the arXiv preprint, 2024.
167. Bharadhwaj, H.; Mottaghi, R.; Gupta, A.; Tulsiani, S. Track2Act: Predicting Point Tracks from Internet Videos Enables Diverse Zero-Shot Robot Manipulation. In Proceedings of the arXiv preprint, 2024.
168. Wu, Y.; et al. Video Prediction Policy: A Generalist Robot Policy with Predictive Visual Representations. In Proceedings of the Proceedings of the International Conference on Machine Learning (ICML), 2025.
169. ViPRA Team. ViPRA: Video Prediction for Robot Actions. *arXiv preprint arXiv:2511.07732* **2025**.
170. Mimic-Video Team. Mimic-Video: Video-Action Models for Generalizable Robot Control Beyond VLAs. *arXiv preprint arXiv:2512.15692* **2025**.
171. FOFPred Team. Future Optical Flow Prediction Improves Robot Control and Video Generation. *arXiv preprint arXiv:2601.10781* **2026**.
172. Meta AI. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning. *arXiv preprint arXiv:2506.09985* **2025**.
173. UP-VLA Team. UP-VLA: A Unified Understanding and Prediction Model for Embodied Agent. *arXiv preprint arXiv:2501.18867* **2025**.
174. NVIDIA. Cosmos: World Foundation Models for Physical AI. *arXiv preprint* **2025**.
175. Nasiriany, S.; Xia, F.; Yu, W.; Xiao, T.; Liang, J.; Dasgupta, I.; Xie, A.; Driess, D.; Wahid, A.; Xu, Z.; et al. PIVOT: Iterative Visual Prompting Elicits Actionable Knowledge for VLMs. *arXiv preprint* **2024**.
176. Wu, J.; Antonova, R.; Kan, A.; Lepert, M.; Zeng, A.; Song, S.; Bohg, J.; Rusinkiewicz, S.; Funkhouser, T. TidyBot: Personalized Robot Assistance with Large Language Models. In Proceedings of the Autonomous Robots, 2023.
177. Liu, Z.; Chi, C.; Cousineau, E.; Kuppaswamy, N.; Burchfiel, B.; Song, S. ManiWAV: Learning Robot Manipulation from In-the-Wild Audio-Visual Data. *arXiv preprint* **2025**.
178. Google DeepMind Gemini Robotics Team. Gemini Robotics: Bringing AI into the Physical World. *arXiv preprint arXiv:2503.20020* **2025**.
179. NVIDIA Robotics Team. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. *arXiv preprint arXiv:2503.14734* **2025**.
180. Figure AI. Helix: A Vision-Language-Action Model for Humanoid Control. <https://www.figure.ai/news/helix>, 2025.
181. 1X Technologies. 1X World Model. <https://www.1x.tech/discover/1x-world-model>, 2025.
182. Dyna Robotics. DYNA-1: The First Commercial-Ready Robot Foundation Model. <https://www.dyna.co/dyna-1/research>, 2025.
183. Ma, Y.J.; Liang, W.; Wang, G.; Huang, D.A.; Bastani, O.; Jayaraman, D.; Zhu, Y.; Fan, L.; Anandkumar, A. Eureka: Human-Level Reward Design via Coding Large Language Models. *International Conference on Learning Representations (ICLR)* **2024**.

184. Sunday Robotics. No Robot Data: How ACT-1 Learns from Human Demonstrations. <https://www.sunday.ai/journal/no-robot-data>, 2026.
185. Covariant. Introducing RFM-1: Giving Robots Human-Like Reasoning Capabilities. <https://covariant.ai/insights/introducing-rfm-1-giving-robots-human-like-reasoning-capabilities/>, 2024.
186. Boston Dynamics.; Toyota Research Institute. Large Behavior Models and Atlas Find New Footing. <https://bostondynamics.com/blog/large-behavior-models-atlas-find-new-footing/>, 2025.
187. Tesla AI. Tesla Optimus: A General-Purpose Humanoid Robot, 2025. <https://www.tesla.com/optimus>.

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