Vision-Language-Action (VLA) Models for Bimanual Manipulation and Their Real-World Deployment: A Comprehensive Survey

Inkyu Sa
inkyu@chefrobotics.ai

***Abstract*—Vision-Language-Action (VLA) models place visual perception, instruction understanding, and motor control inside a single learned policy. This survey reviews that literature through the lens of *bimanual manipulation*: coordinated two-arm control, which remains among the most demanding open problems in embodied intelligence and which stresses every component of a VLA at once. Covering more than 200 cited sources, we organize the field along seven dimensions: architectural foundations (autoregressive, flow-based, diffusion, hybrid), training recipes, action representations, coordination strategies, language grounding, cross-cutting concerns including memory and predictive world models, and real-world deployment. Our organizing claim is that *coupling tightness*, rather than task identity, predicts which architectures succeed: loosely coupled tasks yield to decomposition, whereas tightly coupled tasks require joint generation that preserves inter-arm correlation. We include the newest generation of systems ($\pi_{0.7}$, Gemini Robotics On-Device 2, GR00T N1.7, Xiaomi-Robotics-0) alongside comparison tables spanning 31 methods, and we survey commercial deployments across food assembly, manufacturing, logistics, household, surgical, laboratory, and agricultural settings. The strongest reported two-armed results come from flow-matching heads with action chunking, and we give the mechanical reason: near-straight transport paths permit roughly ten integration steps, which is what fits a generative head inside a real-time budget. We are careful about how far that evidence reaches. The systems reporting tightly coupled dual-arm results are largely from a single family, their protocols are not shared, and no standard bimanual benchmark exists, so the comparison rests on self-reported results under differing conditions rather than on a controlled measurement. Three gaps persist. Dexterity and force sensing remain absent from systems that manipulate under sustained contact; evaluation is the field's weakest instrument, since no shared bimanual benchmark exists and typical trial counts cannot resolve the differences claimed; and deployed scale has come apart from demonstrated capability, with the largest fielded systems remaining narrow and the reliability required in production exceeding published benchmark rates by a wide margin. We close with three research directions addressing these gaps.**

***Index Terms*—Vision-Language-Action models, bimanual manipulation, robot learning, imitation learning, flow matching, action chunking, foundation models, embodied intelligence**

I. Introduction

Bimanual manipulation, the coordinated use of two robotic arms to perform tasks such as folding, assembly, and object reorientation, is fundamental to performing useful work in human environments. From folding laundry to clearing a dinner table, the tasks that matter most in homes, hospitals, and factories often require two-handed dexterity. Fig. 1 shows learned manipulation already running in four such settings, and the distance between those scenes and the benchmark numbers this survey tabulates is one of its recurring themes.

Fig. 1: Learned manipulation in four deployment settings. These are the organizations' own published images and do not all correspond to the specific programs tabulated in Table XIII. **Figure**: two humanoids spreading a duvet, coordinated multi-robot, not single-robot bimanual, manipulation. **Chef Robotics**: hygienically sheathed arms portioning onto a moving conveyor, operators alongside. **Physical Intelligence**: a rail-mounted arm in an ordinary kitchen. **Boston Dynamics and Google DeepMind**: whole-body reaching at a parts rack on the Atlas platform. Panels reproduced from each organization's public materials; copyright remains theirs.

Despite decades of research in motion planning and control, programming a bimanual robot to perform even a single multi-step task remains prohibitively expensive in engineering effort. The core difficulty lies in the high-dimensional, contact-rich nature of bimanual coordination: two arms with 7+ degrees of freedom each must move in concert, often under partial observability and with deformable or articulated objects.

The rise of *Vision-Language-Action* (VLA) models offers a fundamentally different path. By drawing on the representational power of large vision-language models (VLMs) pre-trained on internet-scale data, VLAs learn visuomotor policies that map raw image observations and natural-language instructions directly to robot actions. Pioneering systems such as **RT-2** [1] demonstrated that a single VLM backbone can be fine-tuned to output robot actions, inheriting broad semantic understanding from web-scale pre-training. Subsequent work has rapidly expanded the VLA family: π_0 [2] introduced flow matching for continuous action generation, **OpenVLA** [3] provided the first fully open-source VLA, and $\pi_{0.5}$ [4] scaled

VLA deployment to open-world household tasks with a fleet of mobile manipulators.

The pace of progress has been striking. In 2022, **RT-1** [5] showed that Transformer-based policies trained on large-scale robot data could generalize across tasks and environments. By mid-2024, π_0 [2] achieved state-of-the-art performance on bimanual dexterous tasks (folding laundry, assembling boxes, busing tables) by combining a pre-trained VLM with a flow-matching action head. In early 2025, $\pi_{0.5}$ [4] extended this to autonomous operation in novel homes, while π_0^* [6] introduced RECAP, so that VLAs can improve from their own autonomous experience via reinforcement learning. Concurrently, diffusion-based approaches such as **RDT-1B** [7] scaled to 1.2 billion parameters for bimanual diffusion foundations. **TinyVLA** [8] attacked the inference-latency bottleneck that had limited real-time bimanual control, while **FAST** [9] attacked the training cost of autoregressive policies by compressing the action representation. In 2026 the frontier moved again: $\pi_{0.7}$ [10] showed that a single generalist model can be steered to specialist-level bimanual behavior through prompt-level conditioning rather than per-task fine-tuning, and folded real-time chunking into training; **Gemini Robotics On-Device 2** [11] adapted to an unseen bi-arm platform in hours from fewer than 200 examples while running without a network connection; and open-weight releases such as **Xiaomi-Robotics-0** [12] and **GR00T N1.7** [13] published complete real-time recipes with per-accelerator latency figures.

Several surveys have reviewed related topics. Surveys on foundation models for robotics [14] cover high-level task planning and reasoning but do not address the low-level motor control required for bimanual coordination. Reviews of imitation learning and diffusion policies [15] address action generation mechanisms in detail but predate the VLA framework and do not consider the unique properties of VLM-based backbones. Surveys on multi-arm robotic systems [16], [17] focus on classical motion planning, force control, and task allocation without treating learning-based approaches. Liu *et al.* [18] provide a thorough survey of deep learning for generic object detection that serves as our methodological reference, but the robot learning domain lacks a comparable treatment of VLA models. What no existing survey does, to our knowledge, is organize this literature around bimanual *coupling tightness* (Section II-D), the property we argue predicts which architectures succeed. Nor does any pair that analysis with the record of real-world deployment (Section XI). Coordination, contact modeling, and action-space dimensionality each behave differently across coupling regimes, and a taxonomy built on model family rather than on coupling obscures that.

The timing of this survey is motivated by two converging trends. First, VLA architectures have matured rapidly: between 2023 and 2025, the field progressed from the first VLA demonstrations (**RT-2** [1]) to autonomous bimanual operation in real homes ($\pi_{0.5}$ [4]) and self-improving systems (π_0^* [6]). Second, bimanual hardware has become widely accessible through platforms such as **ALOHA** [19] and **UMI** [20], creating a growing community of researchers with the means to collect bimanual data and deploy VLA policies. This convergence of capable models and accessible hardware makes a systematic review both timely and useful.

The main contributions of this survey are:

- A taxonomy that organizes VLA architectures by action generation mechanism (autoregressive, flow-based, diffusion-based, hybrid) and cross-cuts it with bimanual *coupling tightness*, which we argue is the property that predicts which architecture succeeds on a given two-arm task.
- Comparison tables covering 31 VLA methods across architectural design, training recipes, action representations, and reported benchmark performance, each cell traceable to the work that reports it.
- A survey of real-world commercial deployments across food assembly, manufacturing, logistics, household, surgical, laboratory, and agricultural settings, with an explicit account of the gap between benchmark and field performance and the three research directions that gap implies.

The remainder of this paper is organized as follows. Section II formalizes the VLA policy and bimanual coordination problem. Section III reviews foundational concepts in vision-language models, imitation learning, and generative modeling. Section IV surveys datasets, benchmarks, and evaluation protocols. Section V presents VLA architectures organized by action generation type. Section VI covers training recipes and data strategies. Section VII analyzes action representations and real-time execution. Section VIII focuses on bimanual manipulation with VLAs. Section IX discusses language grounding, reasoning, and generalization. Section X addresses cross-cutting concerns. Section XI surveys real-world deployments and applications. We conclude in Section XII with a synthesis of findings and research directions.

II. Problem Definition and Scope

This section fixes the objects the survey reasons about and the symbols used for them. Table I groups the symbols by role rather than listing them alphabetically, since the grouping itself carries information: quantities that describe the policy, quantities that describe a chunk of behavior, and quantities that describe a two-armed embodiment behave differently in the analyses that follow.

A. VLA Policy Formulation

What separates a VLA from earlier visuomotor policies is not the mapping it computes but where its representations come from. The mapping itself is unremarkable: a policy π_θ consumes an image observation, a language instruction, and typically a proprioceptive reading, and returns an action,

$$\pi_\theta : \mathcal{O} \times \mathcal{L} \times \mathcal{Q} \to \mathcal{A}. \tag{1}$$

Here $\mathcal{O}$ holds one or more camera images $\mathbf{I}_t \in \mathbb{R}^{H \times W \times 3}$, $\mathcal{L}$ holds instruction strings, and $\mathcal{A}$ is whatever the embodiment accepts. A single arm with a parallel gripper needs $n{+}1$ numbers per step for an n-joint arm; the two-armed case is developed in Section II-D.

The substantive claim is architectural. A VLA routes its inputs through a backbone that was trained on internet-scale image-text data before any robot data was seen, and reads

TABLE I: Symbols used throughout, grouped by what they describe. Chunk-level and embodiment-level quantities are separated because the product of the two, $H \times d_a$, is what determines the difficulty of bimanual action generation.

Symbol	Meaning
The policy and its inputs	
π_θ	policy with parameters θ
$\mathbf{o}_t$, $\mathbf{I}_t$	observation, and the images composing it
ℓ	language instruction
$\mathbf{q}_t$	proprioceptive state
$f_{\text{vis}}, f_{\text{VLM}}, f_{\text{act}}$	encoder, backbone, action head
A chunk of behavior	
$\mathbf{a}_t$	action at a single step
$\mathbf{A}_t$	chunk of H consecutive actions
H	chunk horizon
K	integration or denoising steps per chunk
$\mathbf{v}_\theta$	learned velocity field (flow matching)
ϵ_θ	noise-prediction network (diffusion)
$\mathbf{z}$, α, γ, σ_k	Gaussian noise; diffusion schedule terms
A two-armed embodiment	
d_a	action dimension per step
$\mathbf{a}_t^L, \mathbf{a}_t^R$	left- and right-arm components
d_L, d_R	their dimensions

actions out of that backbone's internal states. Decomposing into an encoder, a backbone, and an action head,

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

where $\text{Tok}(\cdot)$ tokenizes the instruction and the proprioceptive reading is tokenized alongside the visual and language streams. The interesting design question is what f_{act} looks like, because that choice governs whether the policy can express multimodal behavior and how much latency each decision costs. Section V organizes the field by exactly that choice.

B. Action Chunking

A policy that emits one action per forward pass cannot drive a robot faster than it can think. Contemporary VLAs sidestep this by predicting a block of consecutive actions from a single observation,

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

a device introduced with **ACT** [19] and now near-universal. Two distinct benefits are usually conflated. The first is economic: one expensive forward pass is amortized over H control steps, so a policy costing tens or hundreds of milliseconds can still serve a 50 Hz loop. The second is representational: predicting a block forces the model to commit to a temporally coherent plan, whereas a step-wise policy is free to contradict itself between adjacent decisions.

The price is committed open-loop time. Having emitted a chunk, the robot is executing a plan formed from a stale observation, and H therefore sets a floor on how quickly it can react to anything unexpected. Practitioners recover some closed-loop behavior by blending overlapping chunks or by replanning before a chunk is exhausted; Section VII examines those mechanisms and their costs, which for bimanual work are more consequential than for single-arm work because both arms are simultaneously committed.

C. Flow Matching for Action Generation

Because chunks are high-dimensional and their conditional distribution is multimodal, most recent action heads are generative rather than regressive. *Flow matching* [21] has become the dominant choice, and its appeal is that it learns a transport map by plain regression. One trains a time-indexed vector field $\mathbf{v}_\theta(\mathbf{x}, t)$, $t \in [0, 1]$, whose integral curves carry samples of a simple prior p_0 to samples of the data distribution p_1:

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

with ϕ_t the flow map at time t. Rather than solving this transport problem directly, one regresses the field onto straight-line displacements between paired prior and data samples,

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

evaluated at the interpolant $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$. Instantiated for control, $\mathbf{x}_1$ is a ground-truth chunk and $\mathbf{x}_0$ is Gaussian noise, and the backbone's hidden states enter as conditioning. π_0 [2] was the first VLA built this way, and its practical significance is the step count: because the learned paths are close to straight, useful samples arrive after roughly ten integration steps rather than the tens to hundreds that diffusion schedules typically require. That difference is what makes generative action heads viable inside a real-time budget, and Section VII-B quantifies it.

D. Bimanual Coordination

Adding a second arm changes the problem quantitatively and qualitatively. Quantitatively, the per-step action is a concatenation,

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

so two 7-joint arms with grippers give $d_a = 16$, and a chunk at $H = 50$ becomes an 800-dimensional object to be generated coherently in one shot. Generative capacity that suffices for a single arm is not obviously sufficient here.

Qualitatively, the harder issue is that the two arms are coupled through the object they jointly hold, and the tightness of that coupling varies by task. We distinguish three regimes, and the distinction recurs throughout the survey because different architectures succeed in different regimes:

1) **Independent**: the arms pursue separate subtasks with no shared constraint, as when one holds a container while the other fetches items into it.
2) **Loosely coupled**: timing must agree but forces need not, as in a handover where release must follow grasp.
3) **Tightly coupled**: motion and force must agree continuously, as when both arms tension a sheet of fabric and any disagreement wrinkles or drops it.

Only the third regime genuinely stresses joint action generation, and it is the regime in which the deformable-object and contact-rich results of Section VIII are situated.

Two axes organize this survey, and we choose them because they are the supply and the demand of one quantity: inter-arm correlation within the horizon of a single decision. What a tightly coupled task demands is that this correlation be

preserved across that horizon. What an action head supplies is a particular way of emitting the horizon, serialized token by token or jointly in one pass, at a quantization and a step count the head fixes. Task identity does not determine the demand, since folding a stiff towel one arm at a time is loosely coupled while tensioning fabric between two grippers is not, which is why we do not organize by task family. Backbone and training recipe do not determine the supply, since they change what a policy knows rather than how its output is emitted, which is why our column axis is the head rather than the model. Crossing the two therefore asks one question per cell: can this head emit what this regime requires, and at what cost? The claim is falsifiable in the obvious way. An autoregressive head matching a jointly generating one on a tightly coupled task under a shared protocol would refute it, and we note in Section XII-B that this comparison has not yet been run.

E. Scope of This Survey

We include systems that pair a pre-trained vision-language backbone with a mechanism for producing actions, and we read that literature specifically for what it says about two-armed manipulation. Autoregressive, flow-based, diffusion-based, and hybrid action heads are all in scope, through mid-2026. Policies learned from demonstrations or from reinforcement learning are in scope; analytic motion planning, force-control synthesis, and optimization-based dual-arm coordination are not, and readers wanting that literature are better served by the dedicated surveys we cite in Section I. Aerial and legged embodiments appear only where they share a policy or a training recipe with a manipulator.

III. Background

Readers already fluent in vision-language models and imitation learning may proceed to Section IV. This section supplies the prerequisites, emphasizing the specific properties that later sections depend on rather than giving a general tutorial.

A. Vision-Language Models

Everything a VLA knows about the world before it sees a robot comes from here.

The architectural lineage is short. Self-attention [22], introduced for translation, transferred to vision once images were treated as patch sequences in the **Vision Transformer** [23], displacing convolutional backbones such as **ResNet** [24] in most multimodal pipelines. The training methodology has an equally traceable lineage: pre-train-then-fine-tune from **BERT** [25], scaled dramatically by **GPT-3** [26], then refined by instruction tuning with human feedback [27], the last of which is why a VLA can follow a phrasing it was never trained on. **CLIP** [28] contributed the alignment recipe, learning joint visual and textual representations from 400 million pairs and obtaining zero-shot transfer across recognition tasks.

Generative multimodal models followed. **Flamingo** [29] interleaved visual and textual tokens for few-shot visual learning; **GPT-4** [30] demonstrated multimodal reasoning strong enough to raise expectations for embodied use; open-weight language backbones [31], [32] made the ingredients broadly available, and **LLaVA** [33] showed that visual instruction tuning on modest data yields a capable model. On the commercial side **Gemini** [34] and its successor [35] advanced multimodal reasoning further. Two contributions matter most for robotics specifically: **PaLM-E** [36] demonstrated that a very large language model augmented with vision could reason about embodied tasks including manipulation planning, and **PaLIGemma** [37] with **Gemma** [38] supplied efficient open-weight backbones small enough to sit inside a real-time control loop, which is why they, rather than the largest available models, underpin most current VLAs.

The property that makes these useful for manipulation is that the expensive knowledge is already present. A backbone pre-trained on web data recognizes objects, parses spatial language, and carries rough physical expectations, none of which had to be learned from robot trajectories, and none of which robot trajectories contain in usable density.

Converting a vision-language model into a VLA means adding an action modality, and three routes are in use: tokenize actions and reuse the existing language head [1]; attach a separate head reading from hidden states [2]; or keep the model as a planner and let it command a lower-level policy [39]. Each trades differently between exploiting pre-trained structure and accommodating continuous high-frequency output, and Section V is organized around that trade.

B. Imitation Learning

Nearly every VLA is trained by supervised learning on demonstrations, so the pathologies of imitation are the pathologies of VLAs.

The idea is old: **ALVINN** [40] learned to steer from sensor-to-action mappings in a shallow network. Modern vision-based imitation [41], [42] scaled this to deep networks and teleoperated manipulation, and **time-contrastive networks** [43] showed self-supervised video representations improving downstream performance. In its plainest form, *behavioral cloning* minimizes a supervised loss over demonstration pairs,

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

which is trivially scalable and structurally flawed. Because training data comes from the expert's state distribution and deployment visits the policy's own, small errors carry the system into states it was never taught to handle, where errors grow [44]. Chunking [19] helps by reducing the number of independent decisions per episode, shortening the effective horizon over which error compounds.

Two failure modes matter more for two arms than for one. *Multimodality* defeats the squared loss in Equation 7 directly: when several actions are valid, their mean may be invalid, and reaching around an object from either side averages into reaching through it. Two arms multiply the modes, since both can approach from multiple directions and the division of labor between them (which arm grasps, which supports) is itself often ambiguous. This is the whole reason expressive generative heads displaced regression.

Distribution shift also compounds faster with two arms, because the state space is larger and errors couple: a small drift

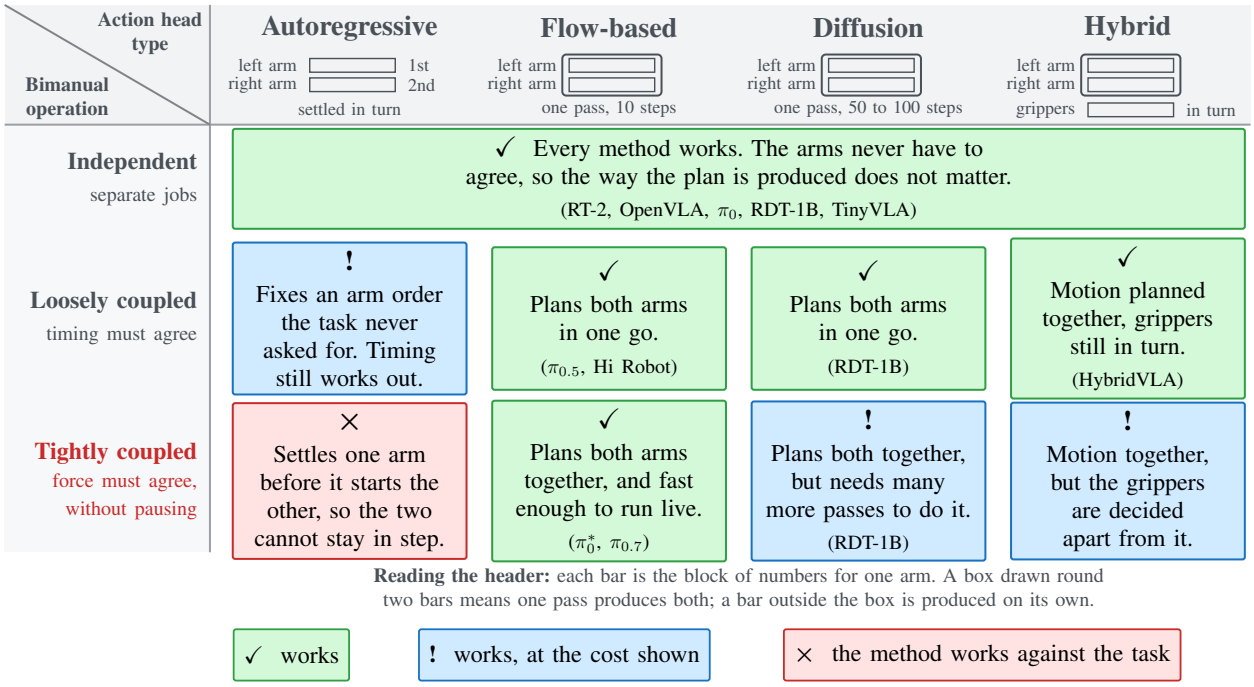

Fig. 2: Which way of producing an action plan suits which degree of two-arm agreement. Rows are the coupling regimes of Section II-D: *independent*, as when one arm holds a container while the other fetches into it; *loosely coupled*, as in a handover where release must follow grasp; and *tightly coupled*, as when both arms tension one sheet of fabric and any disagreement drops it. Columns are the action-head families of Section V: autoregressive (Section V-A), flow-based (Section V-B), diffusion (Section V-C), and hybrid (Section V-D). Flow and diffusion carry the same header picture because they produce both arms the same way, in one pass, and differ only in how many steps that pass costs. Cells state what the mechanism implies, not a measured ranking; the empirical ordering among heads is not established under a shared protocol (Section XII-B).

in one arm's trajectory can make the other's planned motion infeasible, as when a handover target moves. Chunking mitigates without eliminating. The classical remedy of querying an expert during execution [44] is impractical here, since a human cannot supervise two arms in real time without prohibitive cognitive load, which is precisely why autonomous practice (Section VI-C) matters. Adversarial imitation [45] addresses shift through a learned discriminator but introduces training instability and has not been demonstrated at dual-arm action dimensionality.

Language-conditioned imitation [46] extends cloning by conditioning on an instruction, so one policy serves many tasks. VLAs inherit this and improve the encoding, using a pre-trained backbone whose representations generalize across phrasings and compositions the training set never contained.

C. Generative Modeling for Actions

Three generative families have been applied to action decoding, and the relevant history is broader than robotics.

Early latent-variable approaches, variational autoencoders [47] and adversarial networks [48], gave way to denoising diffusion [49] and score-based models [50], which offered higher fidelity at the cost of iterative sampling. **Latent diffusion** [51] showed that operating in a learned latent space cuts cost substantially without sacrificing quality, an insight that reappears in latent-space action generation. Two results specifically about action distributions are worth knowing: **implicit behavioral cloning** [52] showed energy-based models capturing multimodality more faithfully than explicit regression, and **Behavior Transformers** [53] clustered action modes before predicting within a cluster, a discrete-continuous hybrid that prefigures current hybrid heads. On the sequence-modeling side, the **Decision Transformer** [54] recast reinforcement learning as sequence prediction and **Gato** [55] handled text, images, and actions in one autoregressive model: an early demonstration that a single sequence model can emit actions at all.

1) Autoregressive Models: Autoregressive decoding factorizes a chunk into a product of conditionals,

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

which is exactly what a language model already does. **RT-2** [1] exploits this, binning each dimension into 256 intervals and generating indices through the existing language head. Nothing new must be trained, which is the appeal; the costs are quantization error and a decode time growing with the

number of dimensions, both of which bind hardest for two arms, as Section VII-A quantifies.

2) Diffusion Models: Diffusion decoding starts from noise and denoises toward a chunk, iterating

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

with ϵ_θ the noise-prediction network, k indexing steps, $\mathbf{z}$ standard Gaussian noise, and α, γ, σ_k set by the schedule. **Diffusion Policy** [56] demonstrated the approach for visuomotor control. It captures multimodality well and produces smooth trajectories; the difficulty is that useful samples classically need $K = 50to100$ steps, each a full network pass, which is where the latency in Section VII-B comes from.

3) Flow Matching: Flow matching [21], formalized in Section II-C, trains against a simpler objective and typically needs fewer steps. **Rectified flow** [57] explains why: straightening transport paths reduces the integration steps required for a given sample quality, and **scaling rectified flow transformers** [58] showed the approach holding at high resolution with transformer backbones. π_0 [2] demonstrated that ten steps suffice for high-quality chunks, supporting 50 Hz dual-arm control. The mechanism is geometric rather than architectural: a straight path is accurately integrated in few steps, a curved one is not, and that difference is what moved the field from diffusion to flow for real-time work.

D. Bimanual Robotic Systems

Progress in dual-arm learning has tracked hardware accessibility closely enough that the hardware deserves its own account.

Early dual-arm work used industrial position-controlled arms, whose stiffness made contact-rich manipulation difficult. Torque-controlled arms such as the Franka Emika made force-sensitive dual-arm work practical but remained expensive. What changed the field's data situation was purpose-built low-cost hardware.

ALOHA [19] pairs two 6-DOF leader arms with two kinematically identical followers, so an operator moves the leaders and the followers mirror them. Kinesthetic correspondence is what makes dexterous dual-arm teleoperation tractable for a human, and the rig costs under $20 000, which is why it has been replicated widely enough to constitute an ecosystem rather than a single laboratory's apparatus. Paired with the ACT policy, it showed that cloning with chunking learns contact-rich tasks such as threading a zip tie or seating a battery: a CVAE encoder models multimodality during training, and a transformer decoder emits chunks of horizon 100 at 50 Hz, with temporal ensembling smoothing the result.

Mobile ALOHA [59] adds a base and, with it, whole-body teleoperation, base, both arms, both grippers under simultaneous human control. Its more consequential contribution was methodological: co-training with large static-ALOHA datasets improved success even though that data came from a different platform, which is the observation that shaped subsequent VLA training recipes (Section VI-B).

UMI [20] removes the robot from data collection altogether, using handheld instrumented grippers with visual-inertial tracking so demonstrations can be recorded anywhere and retargeted afterwards. This addresses a bottleneck that better teleoperation cannot: demonstrations gathered in one laboratory lack the visual and physical variety that robust policies need, and no rig improvement fixes that. Because collection is decoupled from the robot, non-experts can contribute across kitchens, workshops, and offices.

Two implementation details are easy to overlook and expensive to get wrong. **Calibration** matters more for two arms than for one, because a learned coordination pattern encodes an assumed geometry: relative arm pose, camera extrinsics, and encoder offsets. Errors of about a centimeter in position or two degrees in orientation are enough to break policies that otherwise work, since the pattern was learned against different geometry. **UMI** [20] sidesteps this by recording end-effector poses in a consistent world frame and retargeting to the target robot's kinematics at training time.

Action space choice carries a similar hidden cost. Joint-space targets are direct and embodiment-specific; end-effector poses transfer more readily across kinematics but require inverse kinematics that degrades near singularities. π_0 [2] pads its configuration and action vectors to the dimensionality of the largest robot in its training mixture, 18 dimensions covering two 6-DOF arms, two grippers, a mobile base and a vertically actuated torso, so the representation is configuration-space rather than a per-arm end-effector pose. **ACT** [19] and **RDT-1B** [7] likewise command joint targets. The choice follows the deployment intent: joint space when the hardware is fixed, end-effector space when transfer matters. As Section IV discusses next, the standardization of these action spaces is part of what made shared datasets possible at all.

IV. Datasets, Benchmarks, and Evaluation

Progress in this field is measured against a small number of shared datasets and simulators, so it is worth being explicit about what those resources contain and what they leave out. The short answer, developed below, is that the pre-training corpora are large but thin on two-armed data, and the evaluation suites are reproducible but almost entirely single-armed.

A. Pre-training Datasets

Table II summarizes the corpora that current VLAs are pre-trained on. Three account for most of the field's mileage, and they differ in the axis of diversity they optimize.

Open X-Embodiment [60] maximizes *embodiment* diversity: over a million episodes pooled from more than twenty distinct robots contributed by 21 institutions. Its value is that a policy trained on it sees many kinematic configurations and camera placements, which is why it underwrites **OpenVLA** [3], **Octo** [61], and the pre-training mixture of π_0 [2]. Its weakness for present purposes is that dual-arm episodes are a small minority of the total.

DROID [62] makes the opposite bet, holding the robot fixed and maximizing *scene* diversity: 76 000 trajectories over 564 distinct scenes and 86 tasks, gathered by fifty collectors. Including it in a pre-training mixture reliably improves robustness to unfamiliar surroundings, which suggests that visual

variety and embodiment variety contribute separately rather than substituting for one another.

BridgeData V2 [63], extending the original BridgeData [64], contributes 60 096 tabletop trajectories on a single low-cost arm across 24 environments. Its uniformity is the point: because the setup is consistent and the labeling reliable, it functions as a comparison substrate, and a large fraction of published results quote Bridge numbers.

More recently **GigaBrain-0.5M** [65] contributes half a million episodes mixing teleoperation with autonomously collected rollouts, and unlike the three above it deliberately includes dual-arm episodes and is built to support world-model-based training.

The pattern worth extracting is that no existing corpus was designed for bimanual manipulation. Dual-arm data enters incidentally, as a subset of a cross-embodiment pool, which means that a policy's two-armed competence is largely acquired during task-specific fine-tuning rather than during pretraining.

B. Simulation Benchmarks

Simulators supply what physical evaluation cannot: repeatability and volume at negligible marginal cost.

LIBERO [66] has become the default. Its design decision that matters is factorization: rather than one aggregate score, it separates spatial reasoning, novel objects, goal specification, and multi-step execution into distinct suites, so a weak result localizes the weakness. Four evaluation suites of ten tasks each, with fifty demonstrations per task, sit alongside a larger training split.

SIMPLER [67] answers a different question: not "how good is this policy" but "can simulation stand in for a real evaluation". By calibrating physics and appearance against specific physical setups, it reports correlations above $r = 0.8$ between simulated and real success for most task categories, which licenses simulation as a screening tool even where absolute rates do not transfer.

Beyond these, **RLBench** [68] offers 100 procedurally generated language-annotated tasks, **Meta-World** [69] fifty parametric tabletop tasks aimed at multi-task and meta-learning, **RoboSuite** [70] a modular substrate with standardized robot models, **ManiSkill2** [71] soft-body and articulated-object tasks at scale, and **BEHAVIOR-1K** [72] a thousand household activities in realistic interiors. Li *et al.* [73] examine when simulated evaluation predicts physical outcomes, concluding that both physics fidelity and visual realism matter independently.

For a survey concerned with two arms, the salient fact is an absence: neither LIBERO nor SIMPLER includes dual-arm tasks in its standard suites. The reasons are practical (dual-arm physics, sustained multi-contact interaction, and deformable materials are all expensive to simulate faithfully) but the consequence is that the field's most-used benchmarks cannot measure its hardest coordination problems. Papers that need bimanual numbers build private environments, which do not compare across publications.

C. Evaluation Metrics

Task success rate carries most of the weight in this literature: over N evaluation episodes,

$$\mathrm{SR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\text{task}_i \text{ completed}]. \tag{10}$$

Its virtues are interpretability and comparability; its defects are that it discards partial progress, says nothing about how long the task took, and, because N is typically between ten and fifty, carries a confidence interval far wider than the differences it is used to adjudicate.

Normalized score averages across suites of differing size, permitting a single figure per method. **Inference latency** is the wall-clock cost of producing one chunk, and it binds directly: a dual-arm system at $50\,\mathrm{Hz}$ has $20\,\mathrm{ms}$ per control step, or $H \times 20\,\mathrm{ms}$ per chunk, within which everything must complete. **Data efficiency** counts demonstrations needed to reach a target rate, and matters disproportionately for two-armed work because dual-arm teleoperation is the most expensive data to collect.

Generalization is usually expressed as a ratio between novel and training conditions,

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

where unity would indicate transfer without loss. Neither of the works we cite reports this ratio directly; it has to be computed from separately reported novel and training rates, and few papers publish both under one protocol. That is itself the finding: the axis the field most wants to claim progress on is the one it least often measures in a comparable way.

Two-armed evaluation remains the weak point. Task suites are chosen per paper (π_0 [2] reports laundry folding, box assembly, and table busing) and completion criteria vary between binary success, staged progress, and unstated judgment. Temporal efficiency is rarely reported at all. This is why the cross-method comparisons in Section VIII must be read as indicative rather than as measurements, and it is the first item on the research agenda in Section XII.

V. VLA Architectures and Foundations

We classify policies by how they turn backbone activations into motor commands, because that single choice governs expressiveness, latency, and (as Section VIII argues) suitability for coordinated two-arm control. Four families result: autoregressive, flow-based, diffusion-based, and hybrid, joined by a fifth group defined by deployment target rather than by mechanism. Fig. 3 places them chronologically, Fig. 2 shows the survey's organization, Fig. 4 contrasts the mechanisms, and Table III compares representative systems.

A. Autoregressive VLAs

The autoregressive family makes the smallest possible change to a language model: keep the generative machinery and reinterpret its outputs as actions.

TABLE II: Pre-training corpora, characterized by the axis of diversity each maximizes. "Diversity axis" is the dimension along which the corpus is deliberately broad, and it predicts what a policy gains from including it. No corpus was designed with two-armed manipulation as a primary target.

Corpus	Scale	Diversity axis	Dual-arm	Lang.	Collection method
OXE [60]	>1M episodes, 527 tasks	embodiment (22 robots, 21 institutions)	subset	✓	pooled from 21 institutions
DROID [62]	76K traj., 564 scenes	scene (1 robot, 86 tasks)	-	✓	50 collectors, 12 months
BridgeData V2 [63]	60K traj., 24 environments	task, on a uniform rig	-	✓	single-lab teleoperation
ALOHA [19]	∼1K episodes, 6 tasks	dexterity, dual-arm native	✓	-	bilateral leader-follower teleop
GigaBrain-0.5M [65]	500K episodes, 200+ tasks	source (real + generated)	✓	✓	teleop + autonomous + world model

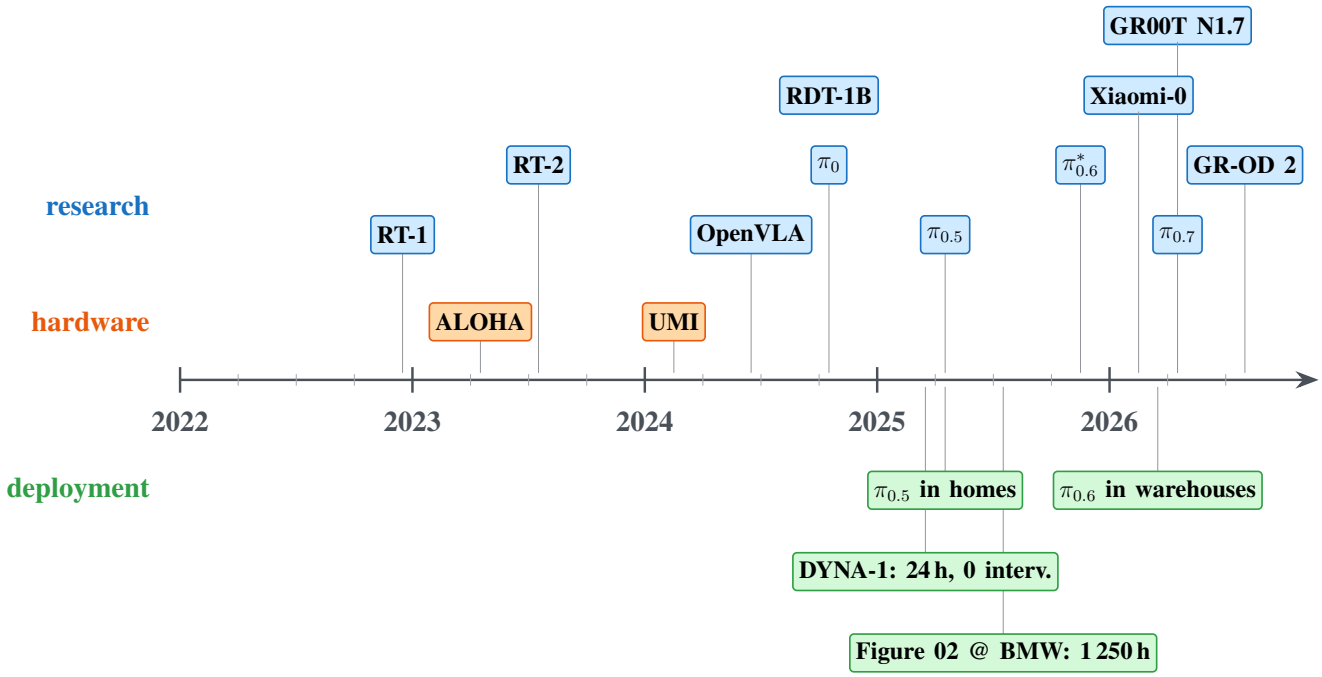

Fig. 3: Milestones on a uniform time axis: research releases (blue), dual-arm hardware and collection interfaces (orange), and the first sustained field deployments (green), each placed at its actual release month. Field operation begins roughly two years after the capability it rests on, and the deployment track is both sparser and narrower in scope (Section XI).

1) RT-1 and RT-2: **RT-1** [5] predates the VLA label but established its data premise. Images pass through a FiLM-conditioned EfficientNet, tokens enter a transformer, and discretized actions emerge from per-dimension classification heads, with a six-frame history and a sentence-encoder language embedding. Trained on 130 000 demonstrations from a mobile-manipulator fleet, it reports 97% on trained tasks and 76% on unseen ones. What carried forward was not the architecture but the finding that heterogeneous data at scale produces generalization: a claim about corpora rather than networks. RT-1 is single-armed and says nothing directly about coordination.

RT-2 [1] took the decisive step of using one model for understanding and action together, fine-tuning PaLI-X (55B) or PaLM-E (12B) to emit discretized actions as text tokens. Capabilities appeared that RT-1 lacked entirely: reasoning over object categories, following phrasings absent from robot data, and rudimentary multi-step planning. This established the field's central empirical claim, that semantic knowledge held in backbone weights transfers to manipulation. Its limitations were equally instructive, closed weights, and a parameter count demanding accelerator-class hardware, which places real-time control out of reach and makes independent reproduction impossible.

2) OpenVLA: **OpenVLA** [3] supplied what RT-2 withheld: a fully open 7B policy on the Prismatic backbone, discretizing into 256 bins per dimension and trained on OXE. It matches **RT-2-X**, the cross-embodiment variant reported alongside **RT-1-X** in the Open X-Embodiment study [60], [74], while being freely available, and its release arguably did more for community research than its benchmark numbers did. It also made scaling behavior legible: performance tracks data diversity, and fine-tuning on small task-specific sets substantially beats the pre-trained checkpoint. As Section VII-A notes, OpenVLA is also the field's most useful negative result on latency: fully open, and architecturally unable to serve a real-time dual-arm

loop.

3) Octo: **Octo** [61] sits awkwardly in any taxonomy, and usefully so. It is a purpose-built transformer rather than a fine-tuned vision-language model, accepts either language or goal images, supports flexible action spaces, and decodes through a *diffusion* head. Trained on 800 000 OXE episodes, it serves mainly as an initialization for downstream fine-tuning, including dual-arm ALOHA tasks. Because its head is diffusion-based it straddles two of our categories, which illustrates that backbone and action head are independent design axes: a point a four-family framing can obscure.

Several systems extend this family in specific directions. **GR-1** [75] treats control as video prediction, emitting actions alongside predicted frames so an implicit forward model comes for free, anticipating the world-model work of Section X-D. **HAMSTER** [76] couples reasoning tightly to action prediction in a single autoregressive pass. On efficiency and precision, **SpatialVLA** [77] adds explicit spatial representations to address the metric imprecision of semantic encoders discussed in Section X-A, **BAKU** [78] offers a compact multi-task architecture, **KAT** [79] bridges perception and motion through keypoint-action tokens, and **SimpleVLA** [80] shows that a deliberately minimal design matches elaborate ones when paired with reinforcement fine-tuning: a reminder that architectural complexity and performance are only loosely coupled.

B. Flow-Based VLAs

Flow-based heads generate continuous chunks by integrating a learned field, avoiding discretization entirely. Fig. 4(b) shows the structure: backbone features condition a field that transports noise into a chunk.

1) π_0: π_0 [2] introduced the pattern that now dominates. A 3B PaLIGemma [37] backbone processes images and instruction; its hidden states condition transformer layers attending jointly to those features and to a noisy chunk, integrated over ten steps.

Its dual-arm results, 80% on shirt folding, alongside box assembly and table busing, were the first to suggest that general two-armed competence was within reach. Three properties combine to produce them: continuous output avoids quantization across sixteen dimensions, ten-step integration fits inside a 50 Hz budget, and generating the chunk in one denoising process preserves inter-arm correlation (Section VIII-A). Pre-training followed by fine-tuning proved essential, the pre-trained checkpoint being what allows a task to be learned from relatively few dual-arm demonstrations.

One caveat should temper how this is read. π_0's strongest numbers rest on proprietary multi-task data gathered across a private fleet, and equivalent results from public data alone have not been demonstrated. How much of the performance is architecture and how much is corpus therefore remains undetermined: a question that recurs for every closed system in Table III.

2) $\pi_{0.5}$: $\pi_{0.5}$ [4] adds hierarchy for open-world operation: a high-level model emits subgoal language and a low-level flow policy executes it. This is what lifts the horizon from a single primitive to a multi-minute task such as cleaning a kitchen, and it suits two arms because the upper layer can decompose an instruction into coordinated dual-arm primitives the lower layer already handles.

Deployed on mobile manipulators in unmodified homes, it reports 94% at following verbal instructions, against 83% task success, on in-distribution household tasks: a real advance in open-world operation, to be read with its provenance in mind, since the result is self-reported under author-chosen conditions and has not been independently replicated. Its separation of instruction following (94%) from task success (83%) in-distribution is more informative than either figure alone, distinguishing grounding failures from execution failures.

3) π_0^* and RECAP: $\pi_{0.6}^*$ [6], which we write as its own report does and which earlier drafts of this survey called π_0^*, addresses imitation's structural ceiling: a cloned policy cannot exceed its demonstrations, and dual-arm teleoperation yields mediocre ones. RECAP has the policy practice autonomously while a vision-language model judges outcomes, accumulating successful episodes without any hand-designed reward. Starting from a π_0 checkpoint and alternating collection with optimization, it reports more than doubled throughput and a roughly halved failure rate on its hardest dual-arm tasks. Section VI-C details the procedure; the architectural significance is that it changed which tasks are reachable rather than improving those already solved.

4) $\pi_{0.7}$: $\pi_{0.7}$ [10] is the newest model in this family and shifts the emphasis from architecture to *steerability*. It pairs a Gemma 3 4B backbone with an 860M-parameter flow-matching action expert for roughly 5B parameters total, adds a multi-scale memory video encoder for history (Section VIII-D), and is trained with the knowledge-insulation recipe [81], in which the backbone is supervised with FAST discrete action tokens while the action expert's gradients are blocked from flowing into it. It consumes up to four camera views with six history frames each, emits a 50-step chunk, executes 15 or 25 of those steps before replanning, and runs at 50 Hz.

The distinctive contribution is *diverse context conditioning*: the prompt carries not only what to do but how, through four channels: a subtask instruction, multi-view subgoal images, episode metadata tagging overall speed and quality and the presence of mistakes, and a control-mode identifier. This allows one generalist model to be steered toward specialist behavior without fine-tuning, which matters for bimanual deployment because per-task fine-tuning is the dominant engineering cost. Reported results are strong: laundry folding at 100% success with roughly $1.5\times$ the throughput of RL-trained specialists, espresso preparation and box building near 100%, and, most relevant to cross-embodiment transfer (Section IX-D), zero-shot shirt folding on an unseen bimanual UR5e at 85.6% task progress against 90.9% for experienced human teleoperators on the same unfamiliar rig. Ablations report that removing episode metadata degrades all tasks and that removing autonomous rollout data substantially hurts performance, which reinforces the RECAP finding that autonomous experience is not merely supplementary. $\pi_{0.7}$ also makes real-time chunking a training-time default rather

than an inference-time add-on, being trained against simulated inference delays of up to twelve timesteps (240 ms at 50 Hz). Weights are not released, and the report does not state end-to-end latency, inference hardware, or absolute training-data scale.

C. Diffusion-Based VLAs

Diffusion heads reach a chunk by reversing a noising process, and reached robot control before flow matching did.

1) Diffusion Policy: **Diffusion Policy** [56] established that generative action models substantially outperform deterministic regression whenever demonstrations are multimodal, which they nearly always are, since several valid ways of performing a task average into an invalid one. Convolutional and transformer variants were both evaluated, the latter stronger on complex tasks.

Its limitation is arithmetic. Fifty to a hundred denoising steps, each a full pass through the noise-prediction network, put per-chunk latency near 300 ms, foreclosing 50 Hz dual-arm control without deterministic sampling or distillation. Though it predates the VLA framing, using task-specific encoders rather than a pre-trained backbone, its contributions were absorbed wholesale: chunk-level denoising, classifier-free conditioning, and temporal ensembling all reappear downstream.

2) RDT-1B: **RDT-1B** [7] scales diffusion to 1.2B parameters explicitly for two-armed manipulation, denoising chunks with a diffusion-transformer backbone conditioned on separate vision and language encoders. Pre-trained on multi-robot data and fine-tuned on ALOHA tasks, it outperforms ACT and Diffusion Policy on tightly coupled work such as handovers and collaborative assembly, and shows improved robustness to visual distraction.

Its use of specialist encoders (SigLIP for vision, T5 for language) rather than a unified backbone is a deliberate trade, forgoing joint vision-language pre-training for strong independent representations and recovering performance through scale. For two arms the transformer backbone's tolerance of long sequences matters concretely, since a 64-step chunk over sixteen dimensions is 1 024 elements that must be denoised coherently.

3) CogACT: **CogACT** [82] pairs a vision-language backbone with a diffusion head but inserts a learned intermediate representation rather than conditioning on hidden states directly: action-oriented tokens attend to backbone features and produce conditioning shaped for denoising. The motivation is that features optimized for language prediction are not automatically the right signal for continuous control, and the abstraction reduces the objective conflict between the two.

Related work informs this family without belonging to it. **PerAct** [83] learns language-conditioned policies over 3D voxels, **RVT** [84] achieves comparable spatial precision far more cheaply by rendering multi-view features, **3D Diffusion Policy** [85] denoises over point clouds for viewpoint robustness, and **Transfusion** [86] unifies next-token prediction with diffusion in one transformer: a template for systems that generate language and continuous action jointly.

D. Hybrid and Efficient VLAs

A final research family either combines mechanisms or optimizes for cost.

1) HybridVLA: **HybridVLA** [87] begins from the observation that action components differ in kind: gripper commands are discrete, arm motions continuous. It routes each to the head suited to it (autoregressive for discrete, flow-matching for continuous) inside one model, outperforming either alone on mixed action spaces.

The fit to two-armed work is natural, since dual-arm tasks are full of this mixture: box assembly interleaves continuous positioning with discrete grasp and release. Because both heads condition on the same backbone pass, their outputs are synchronized by construction rather than by coordination logic. A secondary benefit is training stability, the autoregressive head reusing the backbone's token-prediction competence while the flow head learns continuous output without competing for the same parameters. The cost is tuning: balancing two heads requires care, and no published ablation isolates each head's independent contribution.

2) TinyVLA: **TinyVLA** [8] attacks latency by distillation, training a compact student from a full-size teacher. It reports comparable success at 50 Hz on consumer hardware: the difference between a laboratory result and something a small group can actually run on a dual-arm rig.

3) MiniVLA: **MiniVLA** [88] reaches a similar place by design rather than compression, pairing a small backbone with an efficient head and using residual vector quantization for chunks. At roughly a billion parameters, on a 0.5B language backbone, it retains competitive benchmark performance at a fraction of the compute, which matters for two arms specifically, where both arms' commands must be produced within one control period.

4) FAST: **FAST** [9] makes autoregressive generation competitive again by fixing its representation rather than its architecture. A compression-based tokenizer reduces a chunk to a few dozen discrete tokens, so a language backbone can be trained on high-frequency dexterous data without architectural change, and roughly five times faster. Section VII-A treats the mechanism; the architectural point is that it reaches state-of-the-art autoregressive results while keeping text-token generation, at a per-chunk decode cost that rules it out of a reactive loop.

E. Commercial and On-Device VLAs

A distinct group of recent systems is defined less by its action head than by its deployment target: running on the robot, under a commercial license, at a latency the hardware permits. These matter for bimanual practitioners because they are the models that can actually be obtained and fielded.

Gemini Robotics [89] introduced a cloud-plus-on-robot split, distilling a large backbone so that a local decoder sustains 50 Hz effective control with a backbone latency under 160 ms and roughly 250 ms end to end. It was developed on a bimanual ALOHA-2 fleet and reports new tasks learned from at most 100 demonstrations. **Gemini Robotics 1.5** [90] added embodied reasoning and *motion transfer* across embodiments,

and notably reports that more than 90% of its development evaluation episodes were run in simulation.

Gemini Robotics On-Device [91] removed the network dependency entirely, adapting to new tasks from 50 to 100 demonstrations. Its successor **Gemini Robotics On-Device 2** [11] is built on the Gemma family, is natively multi-embodiment, and adapts to a *new bi-arm embodiment in a few hours from fewer than 200 examples*: a figure of direct consequence for bimanual work, where demonstration collection is the bottleneck (Section VI-D). On two previously unseen platforms it improves markedly over the first generation. Where the first generation moved from 0.0% to 6.7% with adaptation on one bi-arm platform, the second moves from 6.7% to 53.3%; on a second platform the first generation moved from 13.3% to 33.3% and the second from 24.4% to 75.6%. The accompanying **Gemini Robotics 2** model is the first in the line to drive a full humanoid (legs, torso, arms, and multi-finger hands) under one policy [92]. An important caveat for readers tabulating these systems: no technical report accompanies the Gemini Robotics 2 family, so parameter counts, action representation, control frequency, and latency are undisclosed, and the model card is the primary source.

Isaac GR00T N1 [93] is an open-weight dual-system design pairing a vision-language backbone with a diffusion-transformer action head, reported at 63.9 ms on an L40 with System 1 at 120 Hz and System 2 at 10 Hz. The recent **N1.7** revision [13] moves to a commercial license and publishes a full compilation table across accelerators, which is unusual and useful: TensorRT conversion yields 31 ms on an RTX 5090 and 92 ms on a Jetson AGX Thor, against 173 ms on the previous-generation Orin. Those numbers bound what on-robot bimanual inference currently costs.

Xiaomi-Robotics-0 [12] is, to our knowledge, the most completely documented open real-time recipe: a 4.7B model combining a Qwen3-VL backbone with a 16-layer diffusion-transformer flow head, reporting 80 ms on an RTX 4090 with five integration steps on a 30 Hz unified time base, with training-time real-time chunking and open weights and code. On bimanual towel folding it reports 1.2 pieces per minute against 1.0 for $\pi_{0.5}$. **RDT2** [94] scales the diffusion lineage to 8B on more than 10 000 hours of handheld-gripper data and reports the first zero-shot deployment on unseen embodiments at 30 fps within a 16 GB card. **SmolVLA** [95] takes the opposite path, skipping the top half of the backbone's layers to reach 450M parameters at 18 ms per step in under a gigabyte, and is the only model in this group trained entirely on community datasets with transparent provenance.

Among proprietary bimanual systems, **Helix** [96] is notable for its rate separation (a 7B backbone at 7 to 9 Hz driving an 80M visuomotor policy at 200 Hz on a 35-DoF upper body from roughly 500 hours of data) and its successor adds a 10M whole-body controller at 1 kHz [97]. **GR-3** [98] reports that 450 virtual-reality human trajectories, about 30 minutes of collection, suffice for a new bimanual task, and **Redwood** [99] runs a 160M diffusion policy fully on board at $\sim$5 Hz, conditioning on applied forces and training on failure rollouts.

The pattern across this group is a convergence on two-rate architectures, training-time delay conditioning, and open weights for a published baseline while the product model stays closed. Table III places these systems alongside the research models surveyed above.

Table IV compares VLA methods across standard benchmarks. Several patterns emerge. First, the only two methods with traceable LIBERO figures are the OpenVLA pair, and the parallel-decoding variant accounts for almost all the gap between them, which points at the output interface rather than the backbone. Second, hybrid approaches (**HybridVLA**) rank second, suggesting that combining autoregressive and flow-based generation captures complementary strengths. Third, efficient models (**TinyVLA**) trade 10 to 15% performance for significantly reduced compute, a worthwhile tradeoff for resource-constrained deployments.

Autoregressive VLAs benefit most directly from VLM pre-training but struggle with continuous, high-dimensional action spaces. Flow-based VLAs offer the best balance of generation quality and speed for bimanual tasks. Diffusion-based VLAs provide the strongest multimodal modeling but at higher inference cost. Hybrid approaches attempt to combine these strengths, but architecture alone does not determine performance; training strategy plays an equally decisive role.

Two additional modular extensions apply across all four architectural families. First, *memory modules* [102]–[104] augment VLA backbones with explicit short- and long-horizon context, enabling long-horizon task tracking without modifying the core action head (see Section VIII-D). Second, *world models* [105]–[107] complement reactive VLA policies with future state prediction, providing look-ahead planning and synthetic data generation capabilities (see Section X-D).

VI. Training Recipes and Data Strategies

Architecture receives most of the attention in this literature, but the evidence increasingly suggests that training procedure and data composition account for more of the variance in outcome. This section follows a policy through the three stages that produce it (backbone initialization, robot pre-training, task adaptation) then turns to reinforcement learning and to where dual-arm demonstrations come from. Fig. 5 traces the full path.

A. Pre-training

Pre-training happens twice, on two entirely different distributions, and conflating them obscures what each contributes.

The first pass is inherited rather than performed. π_0 [2] begins from PaLIGemma [37], a 3B vision-language model combining a SigLIP encoder with a Gemma [38] language backbone; **OpenVLA** [3] begins from Prismatic, whose 7B configuration pairs DINOv2 and SigLIP encoders with Llama-2; **RT-2** [1] begins from PaLI-X at 55B. What the robot never has to learn, as a result, is what a shirt is, that a handle affords grasping, or that "left of" constrains position.

The choice among backbones is governed less by capability than by whether the resulting policy can run. Larger backbones encode more, but a 55B model cannot serve a 50 Hz loop even with chunking, whereas a 3B model leaves enough budget for a generative head to take several integration steps per chunk. The

TABLE III: Representative systems, ordered by year. "Coupling demonstrated" records the hardest regime (Section II-D) for which two-armed results are reported, which is more informative than a binary bimanual flag: several systems run on dual-arm hardware without demonstrating tight coupling. "n/r" marks a value no primary source reports: the Gemini Robotics 2 family ships without a technical report, so several of its fields are undisclosed.

Method	Yr	Backbone	Head mechanism	Params	Coupling demonstrated	Open
RT-1 [5]	2022	FiLM-EfficientNet	discrete, per-dim. heads	35M	single-arm only	-
Diffusion Policy [56]	2023	task-specific enc.	DDPM, K=50to100	-	tight (short horizon)	✓
RT-2 [1]	2023	PaLI-X / PaLM-E	action-as-text, 256 bins	55B	single-arm only	-
ACT [19]	2023	scratch CVAE	chunked regression, H=100	∼80M	tight, contact-rich	✓
Octo [61]	2024	custom transformer	diffusion head, H=4	93M	independent, loose	✓
OpenVLA [3]	2024	Prismatic	discrete bins, H=1	7B	single-arm only	✓
π_0 [2]	2024	PaLIGemma	flow matching, H=50, K=10	3B	**tight, deformable**	-
RDT-1B [7]	2024	SigLIP + T5	DiT diffusion, H=64	1.2B	tight, handover	✓
TinyVLA [8]	2024	small VLM	distilled diffusion	<1B	independent	✓
MiniVLA [88]	2024	small VLM	residual-VQ chunks	∼1B	independent	✓
$\pi_{0.5}$ [4]	2025	PaLIGemma	hierarchical flow	3B	**tight, long-horizon**	partial
π_0^* [6]	2025	π_0 ckpt	flow + RL (RECAP)	3B	**tight, deformable**	-
Hi Robot [39]	2025	VLM	hierarchical, per-arm subgoals	3B+	loose	-
CogACT [82]	2025	VLM	diffusion via action tokens	7B	independent	✓
HybridVLA [87]	2025	VLM	AR + flow, split by type	7B	tight, rigid	-
FAST [9]	2025	PaLIGemma	DCT + BPE tokens, H=50	3B	loose	✓
SmolVLA [95]	2025	SmolVLM2, layer-skip	flow expert	450M	independent, loose	✓
2026						
$\pi_{0.7}$ [10]	2026	Gemma 3 + video enc.	flow + prompt steering	∼5B	**tight, zero-shot embod.**	-
GR On-Device 2 [11]	2026	on-device Gemma	n/r	n/r	tight (bi-arm, <200 ex.)	-
Gemini Robotics 2 [92]	2026	n/r	n/r	n/r	whole-body + bimanual	-
GR00T N1.7 [13]	2026	Cosmos-Reason2	DiT flow, K=4	3B	tight, rigid	✓
Xiaomi-Robotics-0 [12]	2026	Qwen3-VL-4B	DiT flow, K=5	4.7B	**tight, deformable**	✓
RDT2 [94]	2026	Qwen2.5-VL-7B	RVQ / flow	8B	tight, unseen embod.	✓
Helix 02 [97]	2026	7B VLM (S2)	three-rate hierarchy	7B+	tight, whole-body	-

TABLE IV: Reported evaluation results, restricted to figures we could trace to a stated protocol in the cited source. Cells are left blank where the cited work does not evaluate that benchmark; a blank therefore means *not evaluated*, not zero. LIBERO figures are per-suite values, except where a row spans the four columns to report a suite average. The real-robot column is a single hosted protocol over thirty physical tabletop tasks on four embodiments, which evaluated the five configurations listed and no others.

Method	Simulation (LIBERO)	Real, 30 tasks	Source and protocol
OpenVLA [3]	84.7 / 88.4 / 79.2 / 53.7 (spatial / object / goal / long)	,	as reported in [3]
OpenVLA-OFT [100]	97.1 (suite average)	,	as reported in [100]
Hosted real-robot protocol: 30 tasks, 4 embodiments, 10 rollouts per task [101]			
$\pi_{0.5}$, task-specific	,	**43.7**	best configuration evaluated
π_0, task-specific	,	28.3	
CogACT [82]	,	11.7	
$\pi_{0.5}$, generalist	,	17.7	−26 pts against its task-specific configuration
π_0, generalist	,	9.3	−19 pts against its task-specific configuration

The simulation and real-robot columns use different task sets and protocols and are not comparable as a ratio. Earlier drafts of this table carried LIBERO figures for CogACT, RDT-1B and TinyVLA, and a BridgeData column spanning RT-1 to π_0; those papers do not report LIBERO, and no protocol could be established for the Bridge values, so both have been removed rather than reproduced.

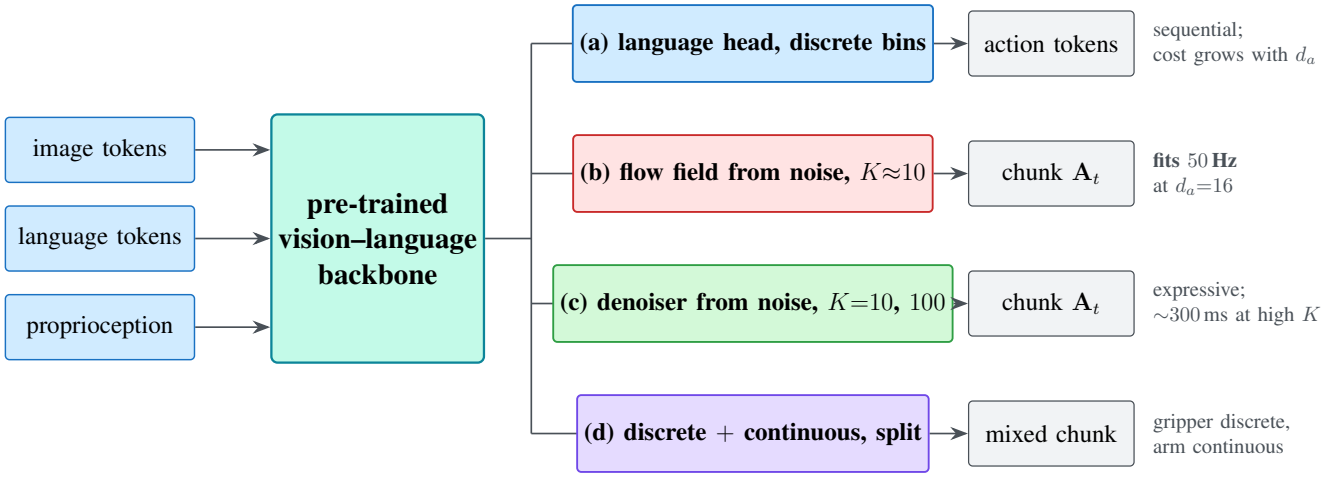

Fig. 4: Four action-head mechanisms sharing one backbone; only the head differs, and each choice sets a latency budget (right). Discrete tokens decode sequentially, so cost grows with action width. A flow field at $K{\approx}10$ fits a 50 Hz dual-arm loop. Diffusion is comparably expressive but slower. A hybrid routes discrete and continuous components separately, both conditioned on the same backbone pass.

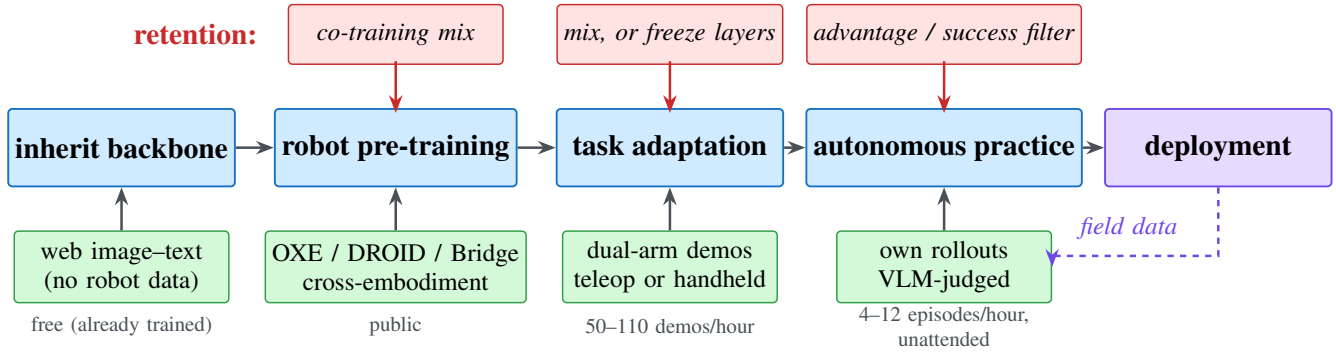

Fig. 5: The training pipeline, showing what feeds each stage (green) and what protects it (red). Data sources differ by orders of magnitude in cost, from inherited web pre-training to dual-arm demonstrations at 50 to 110 per hour. The retention mechanism (data mixing, layer freezing, or outcome filtering) is the substantive difference between published recipes (Table V). Dashed: the deployment flywheel.

field has consequently moved toward smaller backbones paired with more capable heads, the configuration the π_0 family exemplifies, rather than toward scale for its own sake.

The second pass exposes the model to robot data. π_0 trains on OXE together with proprietary data spanning seven embodiments and hundreds of tasks, and does so with a joint objective that preserves language competence while acquiring motor competence:

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

with $\mathcal{L}_{\text{FM}}$ the flow-matching loss of Equation 5, $\mathcal{L}_{\text{LM}}$ the language-modeling loss, and the weights balancing them. Without the second term, fine-tuning on actions degrades exactly the semantic grounding the backbone was chosen for. **OpenVLA** uses only public data and remains competitive, which establishes that proprietary corpora are an advantage rather than a prerequisite.

The consistent finding across this work is that *breadth beats volume*: variety in embodiment, environment, and task predicts downstream generalization better than raw episode count. Two extensions of the principle are worth noting. Memory-augmented policies [102], [103] benefit specifically from pre-training on multi-step tasks, since that is what exercises the memory module. And world models can act as data engines rather than as planners: **GigaWorld-0** [108] generates episodes through video synthesis and sim-to-real transfer, expanding a corpus without collecting anything.

B. Post-training and Fine-tuning

Adaptation to a specific task, robot, or site is where most practical effort goes, and the budgets involved are small. π_0 [2] reaches strong dual-arm performance from 50 to 200 demonstrations per task; **OpenVLA** [3] reports meaningful gains from as few as ten on in-distribution tasks. A well-

initialized policy needs to be told what the task is, not taught to manipulate.

Naive fine-tuning nonetheless degrades what pre-training bought. The standard remedy is to continue mixing pre-training data throughout adaptation, which acts as regularization against overfitting a small target set. π_0 mixes roughly evenly and reports this consistently beating target-only fine-tuning, while **Mobile ALOHA** [59] finds even a tenth of diverse data delivers most of the benefit. The right ratio tracks domain distance: a target task resembling pre-training tolerates aggressive mixing, whereas something structurally unfamiliar, deformable manipulation being the usual example, needs a larger share of target data.

Alternatives to data mixing attack the same problem structurally. **Knowledge Insulation** [81] isolates the backbone from action-expert gradients while supervising it with discrete action tokens, so action learning cannot overwrite language understanding, which preserves instruction following without requiring a mixed corpus at all. Parameter-efficient adaptation via LoRA [109] injects low-rank updates into frozen weights, cutting memory and incidentally limiting drift. Preference optimization such as DPO [110] has been explored for aligning outputs with human judgment, though dual-arm applications remain thin. **Align-then-Steer** [111] separates representation alignment from behavioral steering into two phases, and **Robot Fine-Tuning** [112] pre-trains rewards alongside policies so that autonomous real-world improvement needs minimal supervision. Two systematic studies remain worth reading for their negative results: **RoboMimic** [113] finds observation representation and data quality dominating algorithm choice, and **RoboAgent** [114] pairs semantic augmentation with chunking for generalization.

C. Reinforcement Learning for VLAs

Imitation cannot exceed its demonstrations, and for two-armed tasks the demonstrations are mediocre, teleoperating two arms is slow, cautious, and cognitively taxing, so the ceiling is low. Reinforcement learning is the only mechanism that raises it.

The groundwork predates VLAs. **QT-Opt** [115] demonstrated real-world RL at scale with over half a million physical grasps. Offline methods including **CQL** [116] and **AWR** [117] learn from fixed data without further interaction, avoiding the risk of exploration on hardware; Levine *et al.* [118] survey that setting. **PPO** [119] remains the default for online fine-tuning.

RECAP, introduced with π^*_0 [6], made the loop practical by removing the reward-engineering bottleneck. The policy executes autonomously, a vision-language model judges each episode, successful episodes join the training set, and the policy is refit, then the cycle repeats. Algorithm 1 states it. Using a separate model as the reward function is the load-bearing idea: hand-designed rewards for tasks like folding are impractical to specify, whereas judging whether a shirt ended up folded is exactly what a vision-language model does well. The reported effect is stated by its authors in throughput rather than success terms: on the hardest tasks RECAP more than doubles task throughput and roughly halves the task failure rate. That is a change in what the system can be used for rather than an incremental gain.

Algorithm 1 RECAP: improving a policy from its own experience

Require: policy π_θ (imitation-pre-trained), VLM judge $\mathcal{V}$, tasks $\mathcal{T}$
Require: demonstrations $\mathcal{D}_{\text{demo}}$; practice buffer $\mathcal{D}_{\text{auto}} \leftarrow \emptyset$
for cycle $c = 1 \dots C$ **do**
  for each task $\tau \in \mathcal{T}$ **do**
    roll out π_θ on τ, recording trajectory ξ
    $r \leftarrow \mathcal{V}(\xi, \tau)$ *// no hand-designed reward*
    if r indicates success **then**
      $\mathcal{D}_{\text{auto}} \leftarrow \mathcal{D}_{\text{auto}} \cup \{\xi\}$
    end if
  end for
  refit π_θ on $\mathcal{D}_{\text{demo}} \cup \mathcal{D}_{\text{auto}}$
end for
return π_θ

Other work targets different limitations of imitation. **SAIL** [120] optimizes for speed rather than success, training against time-warped demonstrations so the policy completes dual-arm tasks faster than the human who taught it, worth noting because teleoperation slowness is an artifact of the interface, not a property of the task. **VLA-RL** [121] applies policy gradients directly to the action head, while **ConRFT** [122] adds consistency regularization to keep RL from destroying pre-trained competence. **Q-Transformer** [123] makes offline Q-learning tractable at high action dimension by factorizing the value function autoregressively, and **DPPO** [124] differentiates through a denoising chain to optimize diffusion policies. **Self-improving foundation models** [125] study autonomous generation and filtering of a model's own training data.

GigaBrain-0.5M [65] substitutes imagination for execution, learning a world model that predicts observations under hypothetical actions so the policy can be improved without moving. For two arms this is demanding, the model must predict both arms and the manipulated object evolving together, including contact transitions, and results remain early, but the appeal is that sample cost decouples from wall-clock robot time.

D. Data Collection for Bimanual Manipulation

Dual-arm demonstrations are the field's scarcest input, and three collection strategies with different economics have emerged.

Bilateral teleoperation, as in **ALOHA** [19], has the operator move leader arms that the followers mirror. Kinesthetic correspondence makes it the most intuitive interface for dexterous work, and a complete rig costs under $20 000, which is why it has been replicated widely enough to constitute an ecosystem. **Handheld capture**, as in **UMI** [20], discards the robot at collection time: instrumented grippers with visual-inertial tracking record demonstrations anywhere, and trajectories are retargeted to the robot afterwards. This trades some fidelity for enormous gains in environmental variety and lets non-experts contribute. **Autonomous collection**, as in RECAP [6],

removes the human from the loop entirely once the policy is minimally competent, and has the distinctive property that the resulting data lies on the policy's own state distribution, precisely where behavioral cloning is weakest.

Throughput differs sharply and should inform planning. ALOHA teleoperation yields 50 to 100 demonstrations per hour for short primitives of eight to fourteen seconds, degrading with episode length as resets and operator fatigue accumulate. UMI reports roughly 110 per hour, more than triple a spacemouse interface at about 35, while additionally covering more varied scenes. Autonomous collection needs no operator but is slow per episode when tasks run five to fifteen minutes, giving 4 to 12 episodes per hour, offset by running unattended around the clock. The economics therefore favor a staged approach: teleoperate enough to bootstrap competence, then scale through autonomous practice.

Quality is the underrated variable. Operator skill varies, and novice trajectories teach hesitation and jerk that the policy faithfully reproduces; expert idiosyncrasies transfer just as reliably. Language annotation is noisy in both granularity and phrasing, and automated labeling introduces systematic rather than random error. Most consequentially, demonstrations are not interchangeable: episodes containing recovery from near-failure, varied grasp strategies, or unusual object configurations contribute far more to robustness than additional nominal executions. Levine *et al.* [126] established early that continuous fleet operation can generate corpora large enough for robust grasping, a precedent behind current fleet strategies, and **GigaBrain-0.5M** [65] scores and filters episodes with a world model to retain the informative ones.

E. Data Scaling Laws

How performance responds to more data depends entirely on which data. Ha *et al.* [127] report log-linear returns for language-guided skill acquisition, echoing scaling behavior in language modeling, and the VLA evidence separates into three regimes.

Cross-embodiment pre-training scales log-linearly and rewards curation. **OpenVLA** [3] trains on roughly 970 000 filtered OXE trajectories, having removed degenerate episodes and rebalanced embodiments, and finds diversity at least as predictive as volume.

Task-specific fine-tuning rises steeply then saturates, with the knee set by task complexity. π_0 [2] reports simple tasks converging within about five hours of demonstration while multi-stage tasks such as folding or dishwashing need a hundred hours or more. The practical implication is to spend collection effort on the difficult phases of a task rather than uniformly across it.

Autonomous practice behaves differently again, delivering large gains from little data. Across two RECAP iterations, each gathering roughly 300 trajectories on four robots, the reported gain is a more than doubled task throughput and a roughly halved failure rate on the hardest tasks; box assembly consumed 600 autonomous trials plus 360 with expert intervention per iteration [6]. That a few hundred on-policy episodes can outweigh far larger demonstration sets points to distribution rather than quantity as the operative variable: a theme Section XI revisits with field evidence.

Table V lays these recipes side by side. A three-stage convention has clearly settled (inherit a backbone, pre-train broadly, adapt narrowly) with the genuine disagreement now concerning how to prevent adaptation from undoing pre-training, and whether autonomous experience should follow. How the actions produced by these pipelines are represented and delivered is the subject of the next section.

VII. Action Representations and Real-Time Execution

Section V classified policies by how they generate actions. This section examines the actions themselves: how a continuous motor command is encoded so that a large network can emit it, and what has to happen for the resulting stream to arrive on time. Both questions bind harder in the two-armed case, where the per-step vector is twice as wide and both arms are committed simultaneously.

A. Discrete Action Tokenization

If a policy is a language model, the cheapest way to make it emit actions is to make actions look like words.

Uniform binning does this literally. Each dimension is divided into B equal intervals, $B = 256$ in **RT-2** [1] and **OpenVLA** [3], and the resulting indices are generated as tokens. Nothing about the backbone need change, which is the entire appeal. The cost appears twice. Quantization caps precision at roughly $1/B$ of the action range, and generation is sequential, so decoding time grows with dimensionality. A single arm needs eight tokens per step; a dual-arm system needs sixteen, and that sequential dependency becomes the dominant term in the latency budget precisely where the budget is tightest.

FAST [9] attacks both costs at once by compressing the signal rather than by binning it. The action chunk is transformed with a discrete cosine transform, the coefficients are quantized, and the resulting stream is compressed with byte-pair encoding, so the tokens that survive describe the low-frequency structure of the motion rather than arbitrary grid cells. The compression is substantial: on the authors' fourteen-dimensional 50 Hz shirt-folding set, a chunk that naive binning would spread over hundreds of tokens becomes roughly 53, a $13.2\times$ reduction. The gain grows with control frequency and action width, from $1.75\times$ on a low-frequency single-arm set to $13.2\times$ on the bimanual one, so two-armed data is where the scheme pays most. Two consequences follow beyond speed. Every token is informative, unlike uniform binning where most of 256 bins are never visited. And because byte-pair encoding merges frequently co-occurring sequences, recurring dual-arm motion compresses more sharply than one-off motion, though this is a statistical property of the corpus rather than an explicit codebook of coordinated primitives.

Universal action tokenization [128] pushes this further, learning one codebook shared across embodiments so that a single autoregressive policy can drive robots of different morphology. For two-armed work the attraction is a route

TABLE V: Training recipes across representative methods. The rightmost column records the mechanism each system relies on to avoid degrading its pre-trained backbone during adaptation (data mixing, architectural freezing, or nothing at all) since that choice, rather than the optimizer, distinguishes these recipes in practice.

Method	Backbone init.	Robot pre-train	Adaptation	RL	Retention mechanism
RT-2 [1]	PaLI-X (55B)	Google fleet	co-fine-tune with web data	-	joint web/action objective
OpenVLA [3]	Prismatic (7B)	OXE, ∼970K filtered	task-specific, ≥10 demos	-	none stated
Octo [61]	from scratch	OXE (800K)	task-specific	-	n/a (no VLM to preserve)
π_0 [2]	PaLIGemma (3B)	OXE + proprietary, 7 embod.	50 to 200 demos/task	-	∼50:50 data mixing (Eq. 12)
$\pi_{0.5}$ [4]	PaLIGemma (3B)	OXE + fleet	fleet demos, hierarchical	-	data mixing + subtask supervision
π_0^* [6]	π_0 checkpoint	as π_0	autonomous + demos	✓	mixing + VLM-judged filtering
Mobile ALOHA [59]	from scratch	static ALOHA corpus	50 demos/task	-	10% co-training suffices
Knowledge Insul. [81]	VLM	-	task-specific	-	gradient isolation + token supervision
RDT-1B [7]	SigLIP + T5	multi-robot	ALOHA tasks	-	separate encoders, so little to lose
GigaBrain [65]	VLM	GigaBrain-0.5M (+ synthetic)	task-specific	✓	world-model filtering

by which abundant single-arm data might inform dual-arm behavior, if the tokenizer can place related single- and dual-arm motions near each other in code space.

Several adjacent lines matter here. **Consistency models** [129] distill iterative denoising into one forward pass while retaining distributional coverage, a direct attack on generative latency. **ACT** [19] established chunk scheduling and temporal ensembling as the means of suppressing jitter in dual-arm trajectories. **GNFactor** [130] show shared 3D representations improving multi-task prediction, and **RACER** [131] treats recovery as a first-class output by learning language-guided corrective actions.

The choice between discrete and continuous is ultimately about where error is tolerable. With 256 bins over a ± 1 rad/s range, worst-case quantization error per joint is about 0.004 rad/s, negligible in isolation. But a dual-arm chunk contains sixteen dimensions across fifty steps, and if those errors bias a sustained two-arm contact in a consistent direction they accumulate into a dropped or wrinkled object. Continuous heads avoid the issue entirely and pay in integration steps instead.

B. Continuous Action Generation

Continuous heads emit real-valued vectors, trading discretization error for iterative computation. Flow matching (π_0 [2]) and diffusion (**Diffusion Policy** [56], **RDT-1B** [7]), whose architectures Section V covers, both integrate a learned field from noise toward a chunk.

The step count K is the operative parameter, and the two families sit at different points. Flow matching's near-straight transport paths permit $K = 10$ in π_0, which fits a 50 Hz dual-arm loop with room to spare. Diffusion's curved schedules classically need $K = 50 to 100$, reducible to ten or twenty with deterministic samplers at some cost in fidelity. Since each step is a full pass through the head, this difference is the single clearest reason flow-based heads displaced diffusion in real-time bimanual systems, and it is a statement about sampling geometry rather than about model capacity.

C. Action Chunking Strategies

Given chunking (Section II-B), the horizon H must be chosen, and it is a genuine trade rather than a hyperparameter to be tuned upward.

Short horizons ($H = 1, 4$) keep the policy responsive and permit fast reaction to disturbance, at the cost of frequent inference and myopia; **Octo** [61] uses $H = 4$, which suits its lightweight head. Long horizons ($H = 50, 100$) amortize inference and capture extended structure, but commit the robot to open-loop execution, at 50 Hz, π_0's $H = 50$ is a full second of motion decided from one observation. When both arms move in concert, that second contains the entire coordinated pattern, which is exactly why long chunks work well for folding and assembly.

Task phase structure supplies a principle for choosing. Manipulation primitives decompose into approach, grasp, manipulate, and release, and a horizon that ends mid-phase forces the model to blend behaviors that belong to different phases, producing hesitant or inconsistent motion. A horizon spanning at least one complete phase avoids this; horizons spanning several make the prediction problem needlessly hard. Empirically $H = 50$ at 50 Hz sits near the optimum for most bimanual primitives, and the coincidence of one second with one phase is not accidental, human demonstrators segment at roughly that scale.

Temporal ensembling softens the seam between successive chunks by averaging their overlap. With $\mathbf{a}_t^{\text{new}}$ from the current chunk and $\mathbf{a}_t^{\text{old}}$ from its predecessor,

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

with $\lambda \in [0, 1]$ typically decaying across the chunk. The technique is not universally beneficial: the π_0 authors report having tried it and abandoned it because it degraded performance, choosing open-loop chunk execution instead [2]. Averaging two plans is only sound when both are approximately correct, and near a phase boundary they may encode incompatible intentions.

D. Real-Time Chunking

RTC [132] reframes the latency-reactivity trade as a scheduling problem. Instead of alternating between computing and executing, the two overlap: the next chunk is generated while the current one runs, and the policy transitions to it whenever it becomes available. Crucially, the portion of the current chunk already committed is treated as fixed context for the new one, so the two agree where they meet rather than fighting.

The consequence is that coherence and reactivity stop competing. Reaction time falls to roughly the cost of generating one chunk, 50 to 70 ms for a π_0-class model, rather than the full execution time of a chunk, which at $H = 50$ and $50\,\text{Hz}$ is $1\,000\,\text{ms}$. That is better than an order of magnitude, and it is what allows a two-armed system to respond mid-motion when an object slips from one gripper. Detection of such an event, whether from visual change or a proprioceptive anomaly, can trigger the transition immediately rather than waiting for the scheduled boundary.

E. Bidirectional Decoding

BID [133] operates at sampling time rather than by changing the decoder. At each step it draws several candidate chunks from the policy and selects among them on two criteria: *backward coherence*, which favors candidates consistent with decisions already committed, and *forward contrast*, which favors candidates that remain likely under a stronger policy than under a weaker one.

The first criterion is what matters for two arms. Chunked execution fails at the seam, as Section VII-C discusses, and backward coherence directly penalizes a candidate that disagrees with what the arms are already doing, which is the disagreement that breaks a coordinated motion mid-primitive. Because selection happens at inference, no retraining is required, so the method can be applied to a policy that is already deployed. The cost is that the policy must be sampled several times per control step, which competes for exactly the budget Section VII-D is trying to preserve.

F. Training-Time Action Conditioning

TTAC [134] addresses a mismatch that RTC introduces. At deployment the head must produce a chunk consistent with an already-committed prefix, but if training never presented such a prefix, the head has not learned to respect one. TTAC supplies the prefix during training rather than at inference. A delay is sampled at random each step, the leading actions of the ground-truth chunk are fed in un-noised, and the loss is masked so only the remainder is supervised, so the head learns to complete a chunk given a committed prefix. The inpainting guidance that inference-time chunking applies at every denoising step, with its backward pass, is then unnecessary.

The comparison the source draws is against inference-time chunking rather than against an unconditioned baseline, and it reports no percentage gain on bimanual tasks specifically. What it does establish is the operational point: because the mechanism is a training procedure, it costs nothing at run time, which distinguishes it from methods that buy consistency with extra computation inside the loop.

Table VI collects these choices. Read as a whole it argues against a single recommendation: discrete tokens remain the cheapest way to reuse a language backbone, continuous heads dominate where precision matters, and the scheduling methods are orthogonal to both. What the two-armed setting changes is the weighting, wider action vectors penalize sequential decoding, and simultaneous commitment of both arms raises the value of anything that shortens reaction time.

VIII. Bimanual Manipulation with VLAs

This is the survey's focal section. Two-armed manipulation has been studied from several angles (Smith *et al.* [17] survey the classical control and planning literature, Grannen *et al.* [135] formalize an asymmetric division in which one arm stabilizes while the other acts, and Chitnis *et al.* [136] learn task schemas for efficient dual-arm planning) and we read that literature through the coupling taxonomy of Section II-D. The organizing claim is that coupling tightness, not task identity, predicts which architectures succeed. Table VII summarizes the strategies, Table VIII the reported outcomes, and Table IX the platforms.

A. Coordination Strategies

1) Joint Action Space: The prevailing approach declines to treat the arms as separate agents at all. A single policy emits $\mathbf{A}_t \in \mathbb{R}^{H \times (d_L + d_R)}$, and whatever coordination the task requires is learned implicitly from data. π_0 [2] and **RDT-1B** [7] both do this. The appeal is that no coordination machinery has to be designed, and the model is free to discover patterns a designer would not have specified; the cost is a generation problem of 800 dimensions or more, which demands an expressive head.

Why generative heads suit this particularly is worth making precise. A flow or diffusion head denoises the entire chunk as one object, so the velocity field $\mathbf{v}_\theta$ operates over the full $H \times (d_L + d_R)$ space and every intermediate state is a jointly consistent pair of trajectories. Correlations between the arms are therefore maintained at every timestep by construction, not assembled afterwards.

Autoregressive generation over a concatenated vector has a structural awkwardness here. Some ordering must be imposed, so one arm's actions are predicted conditioned on the other's, introducing an asymmetry the task does not possess. Randomly swapping arm labels during training mitigates the symptom, but the model is still factorizing a joint distribution in an arbitrary direction, and for tightly coupled tasks that factorization is precisely what one wants to avoid.

2) Independent Policies: The alternative decomposes: separate low-level policies per arm, coordinated by a planner above them. **Hi Robot** [39] instantiates this, with a reasoning model issuing per-arm subtask descriptions that independent executors carry out. Each policy then faces a smaller action space, and the decomposition is legible: a genuine advantage for debugging and for safety review.

TABLE VI: Action representations, framed as trades rather than as a ranking. Latency is per chunk on a single GPU as reported by each source. The final two columns state what each choice buys and what it gives up, since none dominates: the appropriate choice depends on whether precision, latency, or reactivity is the binding constraint.

Method	Output	H	K	Latency	Buys	Gives up
RT-2 [1]	256 uniform bins	1	-	$\sim$1 s	backbone reuse with no new machinery	precision, and latency scaling with d_a
OpenVLA [3]	256 uniform bins	1	-	$\sim$150 ms	full openness and reproducibility	real-time viability at dual-arm width
FAST [9]	DCT + BPE tokens	50	-	$\sim$750 ms	13.2$\times$ fewer tokens than uniform binning on bimanual data; 5$\times$ faster training	$\sim$53 sequential decode steps per chunk
π_0 [2]	flow matching	50	10	$\sim$70 ms	no quantization; smooth 16-D chunks	K forward passes per chunk
Diff. Policy [56]	DDPM / DDIM	16	50 to 100	$\sim$300 ms	strongest multimodal coverage	real-time control at 50 Hz
RDT-1B [7]	DiT diffusion	64	20	$\sim$150 ms	longest horizon among open models	inference cost and memory
RTC [132]	flow + overlap	50	10	$<$50 ms*	coherence and reactivity together	added implementation complexity
BID [133]	sample + select	50	n/a	$\sim$100 ms	closed-loop coherence with no retraining	N-fold sampling cost per step
TTAC [134]	flow + prefix cond.	50	10	as base	prefix consistency at zero runtime cost	retraining with a delay-sampling pipeline

*Effective reaction latency once generation and execution overlap; per-chunk compute is unchanged.

The limitation follows directly from the coupling taxonomy. Loosely coupled tasks decompose cleanly, because agreement on timing is all that is required and language can express timing. Tightly coupled tasks do not, because the information the arms must share is continuous force and motion agreement at control rate, and no subtask string carries it. What a joint policy learns implicitly must here be stated explicitly, and language is the wrong bandwidth for it.

3) Leader-Follower: A middle option assigns asymmetric roles: one arm executes the primary motion while the other maintains a constraint, such as holding an object steady. This reduces effective planning complexity and can be expressed within a joint policy by conditioning one arm's output on the other's predicted trajectory. In practice most joint-space systems appear to discover a soft version of this arrangement without being told to, one arm typically initiates contact while the other provides support, which suggests the decomposition reflects something real about the task structure rather than merely being a convenient prior.

B. Contact-Rich Bimanual Tasks

Tasks in which both arms load the same object simultaneously are the hardest category current systems attempt: inserting a peg held in one hand into a socket held in the other, threading, capping a bottle, or aligning parts that must meet under force.

π_0 [2] demonstrated competence on box assembly, where one arm restrains the box while the other folds flaps against it, and the smoothness of flow-generated chunks matters here because discrete-action policies produce contact force profiles that chatter. **ACT** [19] established the underlying point earlier: single-step prediction yields oscillatory contact, whereas chunked prediction holds contact stable across a primitive. Both observations describe the same mechanism, contact stability requires temporal consistency in the command stream.

The unresolved difficulty is that these systems regulate position while the task constrains force. Most VLAs have no force channel at all, so contact state must be inferred visually: from deformation, from induced motion of the contacted object, from occlusion patterns as a grasp closes. That this works as often as it does is genuinely surprising, and suggests the backbone supplies usable physical priors about how contact looks. But visual inference of force has a ceiling, and it is reached wherever the required precision exceeds what appearance reveals, torque-limited fastening being the clear case, since correct and over-tight look identical.

π_0^* [6] improves force behavior without modeling force, by practicing. Through autonomous trials the policy discovers which approach profiles succeed and which cause slip or damage, encoding force-appropriate behavior as learned motion rather than as explicit control. This is among the stronger arguments for reinforcement learning in this domain: the quantity that matters is unobservable to the policy but its consequences are observable in outcome.

Difficulty scales with the number of simultaneous contacts. Two-point contact, one gripper per arm on a shared object, is handled well. Multi-point contact, both arms wrapping a large or compliant object, imposes force-balance constraints that current systems satisfy inconsistently, and multi-fingered hands would multiply contact count again, which is why Section XII-C treats dexterity and force sensing as one problem.

C. Deformable Object Manipulation

Fabric, rope, dough, and bags defeat the assumptions that make rigid-body manipulation tractable: state is high-dimensional, partially observable, and altered by every contact.

Laundry folding is the field's canonical instance, and π_0 [2] reports 80% on folding a shirt. Two ingredients carry that result. Long chunks ($H = 50$, per Section VII-C) let a complete fold be emitted as one continuous trajectory rather than assembled from steps that might disagree; and the backbone

supplies enough visual parsing to identify a sleeve, a hem, or a corner in a crumpled configuration where geometric methods would need explicit state estimation.

$\pi_{0.5}$ [4] generalizes this to arbitrary garments in unfamiliar homes by decomposing hierarchically, the reasoning layer determines which corner to grasp and where to bring it, the motor layer executes, so garment topology is handled symbolically while manipulation stays continuous. $\pi_{0.7}$ [10] subsequently reports folding at 100% success with roughly 1.5 times the throughput of reinforcement-learned specialists, and folds a shirt on a previously unseen dual-arm platform at 85.6% task progress against 90.9% for experienced human teleoperators.

Beyond fabric the category spans rope knotting, dough shaping, and bag opening, and these are not equally hard. One-dimensional deformation admits fairly predictable dynamics; two-dimensional deformation adds self-occlusion, so the state is partly hidden by the object itself; volumetric materials require reasoning about internal structure that no current system handles. The common thread is that a second arm supplies the extra constraint needed to make deformable state controllable at all, one arm holds or tensions while the other shapes, which is why deformable manipulation is intrinsically rather than incidentally bimanual.

Current practice relies on chunks long enough to span a whole primitive, which works when deformation is predictable from the initial configuration and fails when it is not: thin fabric that wrinkles unpredictably mid-fold invalidates a plan formed a second earlier. Tactile sensing to detect material state during manipulation is the obvious remedy and, as Section XI-A notes, is unavailable in exactly the commercial setting where deformable food handling matters most.

D. Long-Horizon Bimanual Tasks

Clearing a table means stacking plates, wiping, and binning items in an order that task semantics dictate, ten to twenty dual-arm primitives over several minutes, where an early error propagates.

$\pi_{0.5}$ [4] maintains a plan in the reasoning layer, issues subgoals, and re-plans on visual evidence, which supplies recovery as a byproduct of the architecture. **Hi Robot** [39] decomposes open-ended instructions the same way. The conceptual lineage runs through language-model planning: **Code as Policies** [137] generates executable programs sequencing primitives, and **SayCan** [138] grounds proposals in learned affordances so that only feasible actions are selected. Both separate reasoning from execution, which is the principle hierarchical VLAs adopt.

A parallel line learns long-horizon structure from unstructured human data. **MimicPlay** [139] extracts plan representations from human play video to guide a low-level controller without step annotations; **PlayFusion** [140] acquires skills by diffusion over language-annotated play; Cui *et al.* [141] study converting uncurated play into conditional policies. Generative and reward-based approaches contribute further mechanisms: **Du *et al.*** [142] use text-guided video generation as intermediate planning targets, **Look Before You Leap** [143] previews consequences with a vision-language model, and **language-image reward models** [144] supply scalable reward signal by aligning descriptions with visual outcomes.

Recovery is what long horizons demand most. When a garment slips mid-fold, the policy must notice, assess, and re-plan. Hierarchical designs get this nearly for free, since the reasoning layer observes the failure and emits a corrective subgoal. Flat policies must learn recovery implicitly from demonstrations that happen to contain perturbations, which is a weak channel. π_0^* [6] improves recovery through autonomous practice, encountering and resolving failures that no demonstrator would have thought to stage.

The horizon also poses a bookkeeping problem. Five minutes at $50\,\mathrm{Hz}$ is $15\,000$ control steps, or 300 chunk-level decisions at $H = 50$. Sustaining coherent task-level intent across 300 decisions requires either hierarchy, where the reasoning layer holds state in its context, or an assumption that each decision is locally determined by the current observation, which is false whenever the task has hidden state, as it does whenever a step has already been completed and must not be repeated.

Explicit memory addresses this directly. **MEM** [102] introduces multi-scale embodied memory, pairing a video encoder for short-horizon visual context, supporting in-context adaptation and recovery from occlusion over seconds, with a language-based memory holding compressed summaries of semantic events over horizons up to fifteen minutes. Integrated into the $\pi_{0.6}$ policy it reports state-of-the-art results on recipe setup, kitchen cleanup, and meal preparation while matching memoryless policies on standard dexterous tasks, and it is reused inside $\pi_{0.7}$ [10]. The design principle is that different timescales want different representations: dense pixels for the recent past, compressed text for the distant past.

Concurrent work explores the same space differently. One group compresses temporal context: **ContextVLA** [104] amortizes multiple frames into a single token, **CronusVLA** [145] extends single-frame policies to multi-frame through learned feature prediction and temporal chunking, **BPP** [146] conditions on model-selected keyframes rather than full histories to suppress spurious correlation, and Torne *et al.* [147] add past-token prediction as an auxiliary objective that regularizes attention to history, reporting threefold gains at a tenth of the training cost. A second group stores and retrieves: **MemoryVLA** [103] maintains a perceptual-cognitive memory bank fusing retrieved context with current observation, **SAM2Act** [148] adapts an episodic-memory segmentation architecture for spatial memory tasks and contributes a benchmark for them, **MemER** [149] retrieves semantically relevant keyframes to guide execution, and **CycleManip** [150] handles cyclic tasks through cost-aware historical sampling.

The benefits are concrete: horizons extending to fifteen minutes, in-context strategy adjustment after an observed failure, tracking of occluded objects and completed steps, and state maintenance across the many subtasks two-armed work involves. Reported gains include $+14.6\%$ on Bridge and $+11.8\%$ on a memory-specific suite over memoryless baselines, and 86.8% across eighteen tasks with only 4.3% degradation under perturbation.

Five caveats temper this. *Causal confusion* is the deepest: with past actions in the observation, the policy can copy its own prior behavior instead of reasoning about the present, because in expert data past and future actions are highly correlated. MEM's language compression partly mitigates this by discarding failed attempts, but the general problem is open. *Train-inference mismatch* follows: training memory holds expert trajectories while deployment memory holds the model's own possibly-erroneous summaries, and errors compound. *Computational overhead* grows with history, and naive multi-frame processing exceeds real-time budgets, which MEM's factorized space-time attention exists to address. *Compression discards* exactly what contact-rich work needs, since language summaries capture semantic events but not force histories. And current memory is *episodic*, resetting between tasks, so nothing accumulates across days of operation. Memory is therefore established as necessary for long-horizon two-armed manipulation while remaining far from the persistent multimodal recall human manipulation relies on.

E. Mobile Bimanual Manipulation

Adding a base makes the workspace a decision variable. The robot must position itself so both arms can reach what the task requires, which couples navigation to manipulation through arm kinematics, task geometry, and obstacles.

Mobile ALOHA [59] demonstrated whole-body teleoperation and imitation for tasks such as cooking and furniture assembly, with chunks spanning base and both arms together. $\pi_{0.5}$ [4] deployed mobile dual-arm robots in unmodified homes, delegating navigation to the reasoning layer's spatial competence while the motor layer handles manipulation: a decomposition that works because positioning tolerances are loose relative to manipulation tolerances.

The two systems illustrate different answers to base placement: Mobile ALOHA learns it implicitly from demonstrations in which the operator positioned well, while $\pi_{0.5}$ reasons about it. **UMI-on-Legs** [151] shows hardware-agnostic collection transferring across mobile morphologies, and **BUMBLE** [152] unifies reasoning and acting for building-scale mobile manipulation. Industrial efforts contribute scale: **AgiBot World** [153] provides a large platform spanning dual-arm and mobile tasks, and **Physical Intelligence** [154] has shown zero-shot foundation-model labeling applied to fleet data. Tasks requiring movement between work areas, fetching from storage, returning to a workstation, remain the hardest case, since they compose long-horizon reasoning with navigation and manipulation simultaneously.

Table VIII is consistent with the coupling hypothesis but does not on its own establish it, since the systems reporting tightly coupled results are largely from one family and no shared protocol exists across them. Loosely coupled tasks are handled well by every method tested, including decomposed ones. Tightly coupled tasks separate the field, and joint-space generative policies lead there despite, more precisely, because of, their high-dimensional output, since a single denoising process preserves the inter-arm correlations that decomposition must reconstruct across an interface. Reinforcement learning from practice improves every row, with the largest gains where force matters and demonstrations are least informative.

TABLE VII: Coordination strategies against the coupling regimes of Section II-D. The rightmost column states what limits each strategy, which is what determines the regime where it fails rather than merely underperforms.

Strategy	**Eff.** d_a	**Regimes served**	**Limiting factor**
Joint space	d_L+d_R	all three	expressive head needed at 800+ dims
Independent	$\max(d_L, d_R)$	indep., loose	language cannot carry force agreement
Leader-follower	d_L	loose, some tight	role assignment must be given or learned
Hierarchical	variable	loose, tight via subgoals	subgoal interface bandwidth

For loosely coupled work, hierarchy buys something joint policies do not: an auditable trace. A subgoal pair such as "left arm: hold the bowl; right arm: stir" is inspectable, which matters for debugging and for the safety arguments Section X-B examines. That interpretability, rather than raw success, is the strongest argument for hierarchical designs in deployed systems.

How humans specify these tasks in the first place, and how robustly policies interpret what they are told, is the subject of the next section.

IX. Language Grounding, Reasoning, and Generalization

Language is the interface that distinguishes a VLA from a visuomotor policy, and it does two separable jobs: it specifies what to do, and it permits correction while doing it. This section examines how instructions reach motor behavior, how hierarchy extends what can be specified, and how far the resulting competence transfers.

A. Language-Conditioned Policies

The pre-VLA approach encoded instruction and image separately and concatenated the results. **Language-conditioned imitation learning** [46] established that this alone permits one policy to serve many tasks; **BC-Z** [155] scaled it toward zero-shot task generalization with a large language-labeled corpus; and Mees *et al.* [156] isolated what drives performance in that setting, finding language-encoder quality and data diversity to be the dominant factors.

VLAs change the fusion point rather than the inputs. Instruction tokens and image tokens pass through shared transformer layers, so the two modalities interact throughout the computation instead of meeting once at the end. The practical consequence appeared in **RT-2** [1], which follows instruction phrasings absent from robot data, handles object categories never manipulated during training, and performs rudimentary spatial reasoning such as placing one object relative to another. None of that was engineered; it is inherited from the backbone and exposed by joint processing.

TABLE VIII: Reported dual-arm results, restricted to tasks the cited work actually evaluates. Each row names its source. Values come from different protocols, task definitions and trial counts and are *not* mutually comparable; they evidence capability on a task, not a ranking between systems. Tasks are grouped by the coupling regime of Section II-D.

Coupling	System	Reported	Task, as evaluated by the source
Tight, deformable	π_0 [2]	80%	shirt folding
	$\pi_{0.5}$ [4]	n/r	garment folding in unseen homes; instruction following 94%, task success 83% in distribution
	$\pi_{0.7}$ [10]	100%	T-shirt and shorts folding, at $\sim$1.5$\times$ the throughput of RL-trained specialists
Tight, rigid	π_0 [2]	n/r	box building, evaluated but not reported as a single rate
	$\pi_{0.7}$ [10]	$\approx$100%	box building
Tight, long-horizon	$\pi^*_{0.6}$ [6]	n/r	laundry in real homes, box assembly, espresso; reported as $>$2$\times$ throughput and $\approx$half the failure rate on the hardest tasks
	$\pi_{0.7}$ [10]	100%	espresso preparation
Contact-rich, fine	**ACT** [19]	80 to 90%	zip-tie threading, battery slotting, cup and tape tasks, from $\sim$50 demonstrations each
	RDT-1B [7]	n/r	ALOHA bimanual suite; reported as outperforming ACT and Diffusion Policy without a shared per-task table
Cross-embodiment transfer			
	$\pi_{0.7}$ [10]	85.6% progress	shirt folding on a previously unseen dual-arm UR5e, against 90.9% for experienced human teleoperators on the same rig

n/r = the source demonstrates the task but does not publish a single comparable success rate. An earlier draft of this table gave four systems a uniform grid of rates across handover, collaborative lift, peg insertion and kitchen cleanup; those tasks are not evaluated in the cited works, and the grid has been replaced with reported results only.

TABLE IX: Dual-arm platforms and collection interfaces. Cost is approximate at introduction. The distinction that matters for data strategy is whether a platform is a robot, a collection device, or both.

Platform	DOF/arm	Cost	Role
ALOHA [19]	6+1	<$20K	robot + teleop rig
Mobile ALOHA [59]	6+1	<$30K	as above, plus a base
Franka dual	7+1	>$60K	robot; torque control
UMI [20]	n/a	<$5K	collection device only

π_0 [2] carries this into the action head, conditioning flow generation on hidden states that already encode both scene and instruction. An abstract directive such as folding something neatly can therefore be grounded in the specific garment present, because the conditioning signal has already resolved what is on the table.

The persistent weakness is brittleness. Minor rephrasing, typographical error, or unfamiliar vocabulary can degrade performance sharply, because the encoder maps such variants to distant embeddings even when intent is unchanged. This is a robustness failure of the language pathway rather than of the motor pathway, and it is under-measured precisely because most evaluations use fixed instruction strings.

B. Hierarchical Reasoning

Instructions worth giving are usually too abstract to execute. **Hi Robot** [39] therefore splits the problem, a reasoning model consumes the open-ended instruction together with the current observation and emits a concrete subgoal, which an executor policy carries out. Making a sandwich becomes opening the bread bag, then retrieving two slices, then placing filling: each within a flat policy's reach.

$\pi_{0.5}$ [4] adds task state to the reasoning layer, so it tracks progress across steps, detects when a subgoal has failed, and re-plans. That capability is what makes minutes-long two-armed tasks feasible, as Section VIII-D discusses.

The intellectual precedent is pre-VLA. **SayCan** [138] paired language-model proposals with learned affordance scores so infeasible suggestions were filtered before execution. Concurrently, **zero-shot planners** [157] showed language models yield actionable structure without robot-specific training, **LLM-Planner** [158] added grounding with environment feedback, **VoxPoser** [159] composed language-model output with vision models into 3D value maps, and **LLM^3** [160] closed the loop from motion failure back to re-planning. Hierarchical VLAs absorb these ideas with the planner and executor now sharing a representational lineage.

What hierarchy introduces is a new design variable: the channel between layers. Its expressiveness bounds how precisely the upper layer can steer the lower one. Natural language, used by **Hi Robot**, is flexible but imprecise about geometry. Code [137] is precise and brittle. Goal images are informationally rich and expensive to produce, though $\pi_{0.7}$'s use of generated subgoal images [10] suggests that cost is falling. Waypoints are geometrically exact and semantically empty. No option dominates, and for tightly coupled two-armed tasks the channel is arguably the binding constraint, since force agreement is not expressible in any of them.

C. Open-Ended Instruction Following

Open-endedness tests whether competence extends past the training distribution. **RT-2** [1] showed it partly does: instructions referring to objects known only from web data were followed, because recognition came from pre-training rather than from demonstrations.

Preserving that ability through action fine-tuning is the difficulty. **Knowledge Insulation** [81] protects it structurally by blocking action-expert gradients from the backbone, retain-

ing instruction-following competence that unconstrained fine-tuning erodes.

Work at the specification extreme includes **Manipulate-Anything** [161], [162], which conditions on instructions detailing not only the goal but the method, and **Stone *et al.*** [163], who use pre-trained vision-language models to reason about objects absent from robot data. **Interactive Language** [164] demonstrates streaming mid-task correction, which is the second of language's two jobs and the one Section XI finds most valuable in the field. **Chain-of-thought predictive control** [165] inserts explicit intermediate reasoning between instruction and action.

The underlying tension is a bandwidth mismatch. Terse instructions are natural for people and leave the policy to supply enormous motor detail; exhaustive instructions remove ambiguity and are unusable in practice. The backbone is what makes the terse end viable, resolving human-level phrasing against visual context. π_0 [2] and $\pi_{0.5}$ [4] both operate from ordinary phrasing, which suggests the bridge holds for two-armed tasks. What is missing is measurement: results are reported on hand-chosen instruction sets, with no protocol for degradation as phrasing departs from training, and no mechanism for detecting instructions that are ambiguous or self-contradictory: a policy asked to stack plates while keeping them separated will do something, and nothing in the architecture flags the contradiction.

D. Cross-Embodiment Transfer

The promise is that data from many robots produces a policy usable on a new one. **Octo** [61] pre-trains across 22 embodiments and adapts with a few hundred target demonstrations; **OpenVLA** [3] reports similar benefit on robots absent from pre-training.

For two-armed systems this matters more than elsewhere, because dual-arm data is the scarcest (Section VI-D). π_0 [2] mixes single-arm and dual-arm sources and finds single-arm data improving dual-arm performance, transfer of visual and semantic representation rather than of motor pattern, since the action spaces differ.

Xiaomi-Robotics-0 [12] illustrates the industrial version, training across platforms with real-time execution as a design constraint and releasing weights and code; its documented latency figures make it unusually useful for practitioners. The broader signal is that emphasis in industrial work has shifted from architecture toward integration, reliability, and inference cost.

Mechanically, reconciling different arm counts and joint configurations is handled three ways: normalizing all embodiments to a shared end-effector representation, supplying an embodiment identifier as input, or learning per-embodiment adapters into a shared space. **Octo** [61] uses embodiment-specific projections, extended in **CrossFormer** [166] beyond manipulation; π_0 pads all embodiments into one fixed-width configuration vector; **RT-H** [167] raises the abstraction level to language-described action primitives that transfer across morphology by not being morphology-specific.

The strongest recent evidence is the zero-shot result in Section VIII-C: $\pi_{0.7}$ [10] folding a shirt on an unseen dual-arm platform at 85.6% task progress against 90.9% for experienced human teleoperators. If it replicates, transfer to new dual-arm hardware is closer to solved than the earlier fine-tuning-based results suggested.

TABLE X: Generalization by axis, annotated with *what supplies it*. The attribution is the useful part: object and instruction generalization arrive with the backbone and therefore saturate early, whereas environment and embodiment generalization must be earned through data and architecture, which is where recent progress has concentrated.

Axis	Current state	What supplies it
Novel objects	largely solved	backbone web pre-training; no robot data needed
Novel instruction phrasing	largely solved, brittle at the margins	backbone language competence; degrades under paraphrase
Novel environment	strong for hierarchical systems	scene-diverse data (DROID) + high-level re-planning
Novel embodiment	recently strong	cross-embodiment corpora, canonical action spaces, and, newly, prompt steering [10]
Novel *coordination pattern*	**open**	not supplied by any current mechanism; must be learned from dual-arm motor experience

E. Zero-Shot and Few-Shot Generalization

OK-Robot [168] composes a vision-language detector with a fixed manipulation primitive to place novel objects in novel homes without task-specific training. It is not a VLA, and that is instructive: much apparent zero-shot capability is perceptual generalization atop a fixed motor skill, which works precisely when the skill is general enough. **Robot Utility Models** [169] pursue the same goal for a broader primitive set.

For two-armed tasks the remaining distance is larger, because coordination patterns are not inferable from language and vision alone; they must be learned from motor experience, and no amount of semantic understanding substitutes.

A methodological caution applies to this whole subsection. Transfer is typically assessed as success after fine-tuning, without controlling how much fine-tuning data was used. A policy needing 200 target demonstrations may be described as transferring, yet the honest comparison is against training from scratch on those same 200, and that baseline is usually absent. Until ablations separate pre-training benefit from fine-tuning benefit, cross-embodiment claims for dual-arm systems should be read as promising rather than established.

Tables X and XI summarize where methods stand.

Two readings of these tables are worth stating. First, instruction and object generalization saturate early, both are largely gifts of the backbone, whereas environment and embodiment generalization improve gradually with data and architecture, which is where recent progress has been. Second, mid-task correction correlates exactly with hierarchy, because accepting

TABLE XI: Language capabilities, separated into specification and correction. The distinction matters because they are supported by different mechanisms and have different deployment value: specification is what the backbone provides, whereas mid-task correction requires an architecture that can accept input while acting.

Method	Hierarch.	Novel phrasing	Novel objects	Mid-task correction	Inter-layer channel
RT-1 [5]	-	limited	limited	-	none (flat)
RT-2 [1]	-	✓	✓	-	none (flat)
OpenVLA [3]	-	✓	✓	-	none (flat)
π_0 [2]	-	✓	✓	-	none (flat)
$\pi_{0.5}$ [4]	✓	✓	✓	✓	natural-language subgoals + task state
$\pi_{0.7}$ [10]	✓	✓	✓	✓	subgoal *images* + metadata + text
Hi Robot [39]	✓	✓	✓	✓	natural-language subgoals
SayCan [138]	✓	✓	-	-	affordance-scored action proposals
Code as Policies [137]	✓	✓	-	-	executable code

new input while acting requires a layer that is not itself in the control loop. That capability is what Section XI finds operators relying on most.

Language and transfer do not act alone: they interact with visual representation, safety machinery, and the practical limits of deployment, which we take up next.

X. Cross-Cutting Concerns

Several issues cut across every architecture and every task category. We treat visual representation, safety, transfer from simulation, predictive world models, and human interaction here, and defer the practicalities of fielding a system to Section XI. Table XII compares how representative methods handle them.

A. Visual Representation Learning

What a policy can do is bounded by what its encoder makes available, and three sources of visual representation are in use.

Backbone encoders, SigLIP within PaLIGemma [37], or a CLIP-style vision transformer [28], arrive already knowing what objects are. That semantic richness is why VLAs generalize to unfamiliar objects. What they are weaker at is metric precision: a representation optimized to distinguish a mug from a bowl need not encode the mug's rim position to millimeters. **Transporter Networks** [170] make the complementary case, showing that equivariant spatial representations learned without large-scale pre-training are highly effective for rearrangement, evidence that spatial structure and semantic breadth are distinct resources rather than points on one axis.

Robot-specific representations target the gap. **R3M** [171] pre-trains on human video with time-contrastive and language-alignment objectives, capturing manipulation-relevant dynamics that image-classification pre-training omits. Nair *et al.* [172] show language-conditioned behavior learned from offline data with crowd-sourced annotation transfers well; **Cortex** [173] finds representations trained on diverse egocentric video outperforming static-image encoders for embodied tasks; and **SPA** [174] makes 3D spatial awareness explicit. Several systems run both pathways, accepting the compute cost to get semantics and geometry together.

Multi-view fusion is not optional for two arms, because no single viewpoint covers both workspaces and the shared object without occlusion. π_0 [2] and **RDT-1B** [7] both take a wrist camera per arm plus one or more third-person views. How to combine them remains open: tokenizing all views and processing them jointly is most common and most expensive; independent encoding with late merging is cheaper and can miss cross-view correspondence; learned view selection reduces cost by attending to whichever camera is task-relevant, which is attractive as view counts grow.

Wrist cameras deserve specific mention because they are what makes fine dual-arm work possible. They supply close-range views of each gripper's immediate surroundings (contact geometry, whether an edge is pinched, whether a grasp has slipped) that no third-person camera resolves. π_0 uses two wrist views plus an overhead; **ALOHA** [19] uses four cameras in total for full coverage. Memory-augmented designs [102], [104], [145] extend representation along time as well as space, which is how occlusion becomes recoverable: what is hidden now was visible a moment ago.

B. Safety

Two arms moving quickly near a person, and near each other, make safety a design requirement rather than a deployment afterthought. Current practice relies on four mechanisms, and it is worth being clear that all four are heuristic.

Bounds and rate limits clip commanded positions and velocities into a safe envelope. They are simple and effective against gross failure, and they do not address the hazard specific to dual-arm systems: the arms can collide with each other while every individual command remains within limits.

Out-of-distribution detection aims to have the policy recognize unfamiliar situations and stop rather than act. Generative heads offer a convenient signal, since spread across sampled chunks indicates disagreement about what to do. For flow-based heads the velocity-field norm at the final integration step is a natural proxy: a large residual means the trajectory has not settled, which is a reasonable reading of low confidence. These signals are useful and uncalibrated: a threshold tuned on one task does not transfer, and no published work reports a false-negative rate.

Collision avoidance between the arms is the dual-arm-specific problem flagged in Section VIII-A. Classical kinematic checking is reliable but external to the policy, which has no notion of its own kinematic constraints. Projecting generated actions onto the nearest collision-free trajectory restores

the guarantee at the cost of latency inside a loop that has none to spare. Learning avoidance implicitly from collision-free demonstrations is cheaper and does not generalize to arm configurations the demonstrations never visited.

Human intervention is the backstop that actually gets used. For deployment in homes, as with $\pi_{0.5}$ [4], the ability to interrupt and redirect verbally matters more than any of the above, and Section XI finds operators relying on it heavily.

Taken together these are mitigations, not assurances. None yields a bound on failure rate, which is why Section XII-C treats safety argumentation, rather than safety mechanism, as the open problem.

C. Sim-to-Real Transfer

Simulation offers unlimited, safe, perfectly labeled data, and the gap between simulated and physical behavior determines how much of that is usable.

SIMPLER [67] takes the pragmatic route of calibrating environments against specific physical setups, reporting correlations strong enough to make simulation a legitimate screening tool even where absolute rates do not transfer.

The gap is worst exactly where two-armed manipulation lives. Friction, compliance, and deformation are the least faithfully simulated phenomena, and they are what tightly coupled dual-arm tasks consist of. Current systems [2], [6] consequently train predominantly on physical data, with simulation in a supporting role: the reverse of the situation in locomotion.

Four strategies narrow it. **Domain randomization** varies simulation parameters during training so the policy is robust across the range the real system might occupy; for two arms, friction, mass, and gripper stiffness are the parameters that matter. **System identification** calibrates the simulator to one specific robot, trading generality for fidelity. Two generative approaches bypass simulation altogether: **Gen2Act** [175] synthesizes human video in novel scenarios to guide manipulation, and **Track2Act** [176] predicts point tracks from internet video to derive policies zero-shot, both treating web video as a physics prior. **Hybrid training** pre-trains in simulation and fine-tunes on hardware, which is standard for single arms and underexplored for two, where higher action dimensionality makes the alignment harder.

D. World Models and Future State Prediction

An alternative to reacting is anticipating: predict the future (as frames, as latent state, or as physical quantity) and act against the prediction. For two-armed work the appeal is threefold: look-ahead over multi-step coordination, the ability to evaluate an action before committing, and synthetic data to relieve the collection bottleneck.

GigaBrain-0 [105] established the pattern, using a generative world model to produce synthetic robot data through video generation and transfer, with depth-aware inputs and embodied chain-of-thought supervision for long-horizon reasoning. Its successor **GigaBrain-0.5M** [65] adds RAMP, which refines world model and policy together through deployment, reporting roughly 30% improvement on dual-arm folding, packing, and beverage preparation. **GigaWorld-0** [108] generalizes the infrastructure with both video and 3D generative components, functioning as a data engine.

A more radical line removes the action head. **Rhoda AI** [106] proposes direct video action, predicting future frames with a causal video model conditioned on history, proprioception, and language, then recovering actions with an inverse dynamics model, reporting high data efficiency at 10 to 20 hours of robot data and native support for hundreds of frames of context. **Video Prediction Policy** [177] learns implicit inverse dynamics through video diffusion, improving 18.6% on one benchmark and 31.6% on dexterous tasks. **ViPRA** [178] learns from actionless video using perceptual and flow-consistency losses at 22 Hz, and **Mimic-Video** [179] pairs internet-scale video models with flow-matching decoders for roughly tenfold sample efficiency.

Prediction need not be pixel-level. **FOFPred** [180] predicts language-conditioned future optical flow, reporting 68.6% on dual-arm tasks against 61.8% for a frame-predicting baseline. **V-JEPA 2** [181] learns a self-supervised world model from over a million hours of internet video, and its action-conditioned variant reaches 65 to 80% zero-shot on novel manipulation, notable because no robot data entered the representation. **WorldVLA** [107] generates actions and future frames in one autoregressive model, **UP-VLA** [182] uses next-frame prediction as auxiliary pre-training, and **Cosmos** [183] supplies open world foundation models trained at very large video scale.

The advantages are real. *Data efficiency*, since web video supplies physical priors that would otherwise require robot time. *Interpretability*, since a predicted frame can be inspected before the robot moves, valuable for debugging and for the safety arguments above. *Planning*, since consequences can be evaluated before commitment, which matters most for two-armed sequences where one wrong move invalidates the rest. And *synthetic data* at scale. Concretely: in a packing task where one arm holds a box open while the other inserts, a world model could predict premature closure and prompt a grip adjustment before contact is lost: a capability wholly absent from reactive policies.

Five limitations bound the paradigm. *Compounding error* is the most serious: autoregressive prediction amplifies early inaccuracies, so long-horizon forecasts become unreliable, and this is worst for contact-rich two-armed tasks where fast contact transitions produce exactly the abrupt state changes video models predict poorly. *Inverse dynamics* adds a second error layer, since recovering precise commands from predicted pixels requires distinguishing action-relevant motion from background change. *Cost* conflicts directly with real-time control: iterative video generation consumes hundreds of milliseconds against a 20 ms per-step budget, and ViPRA's 22 Hz sits below typical dual-arm control rates. *Hallucination* produces plausible but wrong futures, particularly after occlusion, so the robot plans against a world that will not occur. And *no haptic grounding*: these models predict appearance, not force, which is precisely the signal tightly coupled manipulation needs.

E. Human-Robot Interaction

Language makes interaction cheap in a way that teach pendants and demonstration interfaces never did. A user can state a goal, watch, and revise, without stopping the robot or re-recording anything.

$\pi_{0.5}$ [4] demonstrates this conversationally in homes: spoken instructions, observed execution, mid-task redirection, and, because the reasoning layer can generate text as easily as subgoals, clarifying questions when an instruction is ambiguous. **PaLM-E** [36] showed embodied models discussing the physical world, answering questions about object properties, spatial relations, and feasibility, and **PIVOT** [184] elicits actionable guidance through iterative visual prompting.

Evaluation here is the weakest in the survey. Reported results are qualitative or drawn from small user studies, and no standard metric exists for interaction quality, correction latency, or the cognitive load a two-armed system imposes on its supervisor. Since Section XI finds intervention rate to be among the metrics operators care most about, this gap is practical rather than academic.

F. Compute and Communication Constraints

Two engineering constraints shape what is deployable, and both are properties of the loop rather than of the model.

Compute. A 3B-parameter policy taking ten integration steps per chunk needs a high-end accelerator for real-time dual-arm control. Full-size policies do not currently fit comfortably on embedded hardware, which is what motivates the compact architectures of Section V-D and the on-device models of Section V-E. The 2026 releases that publish per-accelerator latency [12], [13] are the most useful reference points, since they let a system designer check feasibility before committing.

Communication. Where inference runs off-robot, transport joins the budget. At 50 Hz the entire path (capture, transfer, inference, return, actuation) must fit inside 20 ms per step. Chunking amortizes the inference term across H steps but sets a floor on reaction time of roughly one chunk period, which is the trade Section VII-D addresses.

A gap between demonstration and dependable operation persists, and it has two components worth separating. Most published results come from controlled conditions with limited environmental variation, and long-run reliability figures, mean time between failures over weeks of continuous operation, are essentially absent from the literature. Separately, reproducibility is limited by the leading systems' reliance on proprietary data and infrastructure. **Xiaomi-Robotics-0** [12] is a partial counterexample, releasing weights, code, and latency measurements for a real-time dual-arm system, and its emphasis on hardware-software co-design rather than architecture is characteristic of industrial work. Related systems illustrate adjacent deployment concerns: **TidyBot** [185] personalizes household behavior to user preference, **Robot Tool Use** [186] studies sensorimotor pre-training for tool manipulation, and **ManiWAV** [187] adds audio, which supplies event signals, a latch clicking, a container filling, that vision misses and that contact-rich two-armed tasks can use.

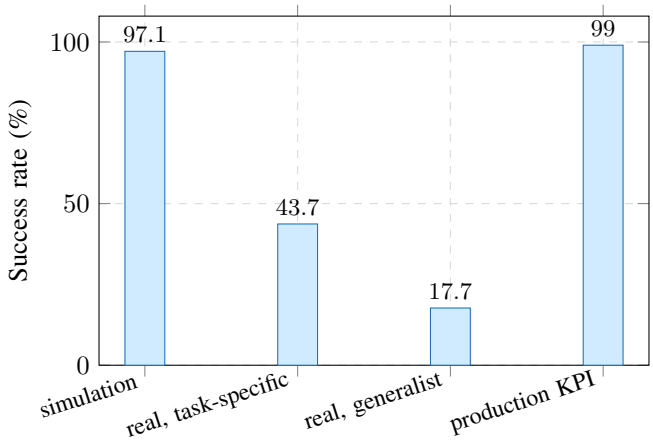

Fig. 6: The same generation of policies, measured four ways. Reported simulation performance [100], best task-specific and generalist results on thirty real tabletop tasks across four embodiments [101], and the placement reliability a commercial automotive cell requires per shift [188]. The three left-hand bars use different protocols and are not strictly comparable; the point is the spread, which spans roughly two orders of magnitude of implied reliability and is not visible from benchmark results alone. Section XI-E develops the implications.

Section XI takes up the deployment question directly, with evidence from systems operating commercially.

XI. Real-World Deployments and Applications

The preceding sections evaluated VLAs largely on benchmarks and controlled experiments. This section turns to systems that have been fielded in commercial settings, because the constraints they reveal differ from those that benchmarks expose and because bimanual manipulation is where several of them operate. Table XIII collects representative examples with their reported scale, and Fig. 1 shows four of these settings, each exhibiting a different constraint profile.

Two qualifications apply throughout. First, essentially all deployment evidence originates from company technical reports, blog posts, and press releases rather than peer-reviewed publications, and *none of the figures below is third-party audited*; every reliability and throughput number is self-reported by the operator. Second, several of the largest deployments are not VLA-based at all, and we include them deliberately, because the comparison between deployed scale and VLA content is itself informative.

A. Food Preparation and Assembly

Food handling has produced the largest reported deployment scale in manipulation. Commercial ingredient portioning has reached 10^8 servings across more than twelve facilities in North America and Europe, handling roughly 2 000 distinct ingredients, with the operator reporting an 88% reduction in food giveaway and a 60% gain in labor productivity [189], [190]. The manipulation is synchronized to a moving conveyor rather than performed in free space, which converts trajectory planning into a timing problem: an instance of a pattern worth

TABLE XII: How representative methods handle the cross-cutting concerns of this section. "Safety" records the specific mechanism rather than a grade, because no method in this table provides a bound on failure rate: the distinction is between kinds of mitigation, as Section X-B argues.

Method	Visual encoder	Views	Proprio.	Safety mechanism	Sim2real	Dialogue
RT-1 [5]	FiLM-EfficientNet	single	-	bounds only	-	-
RT-2 [1]	ViT (PaLI-X)	single	-	bounds only	-	partial
OpenVLA [3]	DINOv2 + SigLIP	single	-	none stated	-	-
Octo [61]	custom ViT	multi	-	none stated	✓	-
π_0 [2]	SigLIP	2 wrist + overhead	✓	rate limiting	-	-
$\pi_{0.5}$ [4]	SigLIP	multi	✓	layered + verbal interrupt	-	✓
π_0^* [6]	SigLIP	multi	✓	rate limiting	-	-
$\pi_{0.7}$ [10]	SigLIP-class + video enc.	up to 4 × 6 frames	✓	rate limiting; delay-robust	-	✓
RDT-1B [7]	SigLIP	multi	✓	bounds only	-	-
Xiaomi-Robotics-0 [12]	Qwen3-VL	multi	✓	unified time base	-	-

noting: fielded systems frequently *reduce* the manipulation problem rather than solving it in full generality.

Three constraints in this domain bear directly on VLA design and are largely absent from the research literature. Sanitation requires washdown-rated hardware, and no arm certifiable for food contact carries exposed tactile sensing, which forecloses the remedy most often recommended for contact-rich manipulation (Section VIII-B). Fig. 1 (top right) shows all three of this domain's constraints at once: the arms are sheathed for washdown and food contact, the deposition is timed to a moving carrier rather than executed in free space, and human operators remain co-located with the cell throughout the shift. And the objects are granular or semi-solid, reflowing after every contact, so their state is not recoverable from a single frame.

Learned bimanual policies have nonetheless reached strong reliability on adjacent tasks. **DYNA-1** [191] reports 99.4% success over more than 24 hours of continuous operation on 850-plus napkins with zero interventions, at roughly 60% of human speed, and a graded-quality breakdown of 98% at three or better on a five-point scale but only 75% at four or five. π_0^* [192] reports an 18-hour continuous espresso-preparation session alongside bimanual box assembly and laundry.

B. Manufacturing and Logistics

Manufacturing supplies the most demanding published specification. A humanoid running the **Helix** policy in an automotive body line was held to a key performance indicator of better than 99% correct placement per shift with zero interventions per shift, a 5 mm tolerance, and 37 seconds of loading inside an 84-second takt, accumulating more than 1 250 hours, 90 000 parts, and 30 000 vehicles over eleven months [188]. That specification differs in kind from a benchmark success rate: takt is a determinism requirement rather than an average, and a bound on interventions per shift is stronger than a bound on failure frequency.

Related industrial work is visible in Fig. 1 (bottom right), where a humanoid crouches at a multi-bin rack to sequence parts; the depth of the crouch is a reminder that such tasks require whole-body control rather than a fixed base with two arms.

In logistics, deployed scale is largest where VLA content is lowest. Parcel sortation reports more than 250 000 production hours and 150 M packages [193]; tactile pick-and-stow more than 500 000 orders [194]; bimanual tote handling more than 100 000 totes under a multi-year service contract [195], [196]; and truck unloading 600 to 800 cases per hour with an explicitly non-learned single-arm perception-and-planning stack [197]. Electronics assembly at 300 000-plus servers uses a hybrid cell in which a human performs prescribed steps inside a sensor-monitored envelope [198]. A cautionary case belongs here too: a multi-arm consolidation cell was withdrawn from operations less than six months after being unveiled, reportedly on cost and manufacturing complexity, without throughput figures ever being published [199].

C. Household and Service Settings

Household deployment is where research results are strongest and commercial reality most qualified. $\pi_{0.5}$ [4] performs long-horizon cleaning in entirely new homes, and the separation it reports between instruction following (94%) and task success (83%) is itself useful, since it isolates language grounding from motor execution. Fig. 1 (bottom left) shows the setting: an ordinary kitchen with unpackaged produce and domestic appliances, the latter relevant because $\pi_{0.7}$ [10] reports acquiring appliance-loading tasks through language coaching alone, without new demonstrations. Commercially, consumer humanoids taking pre-orders have no verified customer deliveries, and publicized demonstrations have been reported as teleoperated, with remote human assistance marketed as a product feature. The honest characterization of the current state is *instrumented partial autonomy*: one bimanual warehouse deployment reports 96.4% autonomy sustained over a full shift at 175 items per hour, and a laundry deployment reports a 50% reduction in teleoperator interventions per load and a 42% reduction in missed grasps after adding site data to pre-training [200].

D. Healthcare, Laboratory, and Agriculture

SRT-H [201] reports 100% success on eight unseen *ex vivo* specimens with no intervention, using corrective language as the error-correction channel: a bimanual surgical result and

one of the clearest demonstrations that language is valuable for *correction* even where it contributes little to within-task control. Autonomous laboratories are quietly in production: one synthesized 41 novel compounds from 58 targets over 17 days of continuous operation [202]. In agriculture, a current-generation VLA fine-tuned for greenhouse harvesting reports 74.0% success at 32.6 seconds per pick with 4.1% damage from only 3.71 hours and 227 teleoperated episodes [203]; the data efficiency is notable and the cycle time remains roughly an order of magnitude too slow to be economic, which identifies seconds-per-unit rather than success rate as the binding metric in that domain. A supermarket restocking study using a current model concluded that bridging the lab-to-store gap is primarily a systems-integration problem [204].

E. What the Deployment Record Suggests

Three observations follow from Table XIII that benchmark results do not supply.

First, there is a persistent gap between benchmark and field performance. On a hosted benchmark of 30 real tabletop tasks across four embodiments, the best task-specific configuration reaches 43.7% success and the same model as a generalist 17.7% [101], against better than 99% per shift in the automotive cell above. Reported success on standard simulation suites exceeds both by a wide margin.

Second, cumulative deployed scale currently correlates *inversely* with how VLA-like a system is. The largest deployments (10^8 servings, 250 000 production hours, 500 000 orders) are narrow and not language-conditioned, while the most VLA-native industrial deployment measures 1 250 hours. This may well be a lagging indicator, since prompt-level steerability [10] and zero-shot cross-embodiment transfer [94] are only months old, but it does argue against assuming that current capability is already deployed at scale.

Third, the metrics that decide deployments are not the metrics papers report. Operators report domain-native units (servings per hour, percentage giveaway, picks per minute, hours of unattended operation, interventions per shift, seconds per pick, damage rate) and graded quality rather than binary success. For bimanual systems approaching saturation on a task, reporting a halving of the failure rate is more informative than reporting 90% becoming 95%.

XII. Discussion and Conclusion

A. State of the Art Performance

Drawing the preceding sections together, the following is our reading of where two-armed VLA manipulation currently stands. Fig. 6 makes the central measurement problem concrete: the same generation of policies reports 97.1% on a standard simulation suite, 43.7% as a task-specific policy on thirty real tabletop tasks, and 17.7% as a generalist on the same set, against a production requirement above 99% per shift.

Architecture. On the evidence collected in Tables IV and VIII, the strongest reported dual-arm results come from flow-matching heads, led by π_0 [2] and its successors $\pi_{0.5}$ [4], π_0^* [6], and $\pi_{0.7}$ [10]. Two properties explain the lead rather than one: continuous output avoids quantization across a sixteen-dimensional action vector, and near-straight transport paths keep the step count low enough to fit a real-time budget. Diffusion heads match the expressiveness, **RDT-1B** [7] shows scale helps, but pay for it in sampling steps. Autoregressive heads integrate most cleanly with a language backbone and are the least suited to wide continuous action spaces, which is a statement about the output interface rather than about the backbone.

Training. The inherit-then-broaden-then-narrow convention is settled. The consequential recent addition is reinforcement learning from the policy's own experience: **RECAP** [6] reports more than doubled task throughput and a roughly halved failure rate on its hardest tasks, which matters more than any architectural increment reported in the same period. Retaining pre-trained competence during adaptation, whether by data mixing or by freezing layers [81], is now understood as a requirement rather than a refinement.

Action representation. Chunking at $H = 50$ is the de facto standard for two arms, supplying roughly one second of coordinated motion per decision. Compression-based tokenization [9] has narrowed the gap between discrete and continuous approaches on task success, compressing a bimanual chunk 13.2× relative to uniform binning and cutting training time roughly 5×, though at ∼750 ms per chunk it remains an order of magnitude too slow for a reactive dual-arm loop, and overlap scheduling [132] with its training-time counterpart [134] has largely dissolved the apparent conflict between chunk coherence and reactivity.

Generalization. Hierarchical designs generalize best to unfamiliar environments and open-ended instructions, with $\pi_{0.5}$ [4] and **Hi Robot** [39] the clearest cases. Cross-embodiment pre-training supplies a usable prior, but two-armed coordination itself still requires task-specific data. The newest result on this axis is qualitatively different: $\pi_{0.7}$ [10] reports steering a single generalist to specialist-level dual-arm behavior through prompt conditioning alone, and folding a shirt on a previously unseen dual-arm platform at 85.6% task progress against 90.9% for experienced human teleoperators on the same unfamiliar rig.

Efficiency. Inference cost remains the constraint that shapes deployable design, and Table XIV summarizes where methods sit. Flow heads at ten steps give the best quality-per-millisecond; compact architectures [8], [88] trade capability for hardware reach; and the 2026 open releases [12], [13] are notable for publishing per-accelerator figures, which earlier work generally did not.

Memory. Explicit memory is the most significant recent addition for long-horizon two-armed work. **MEM** [102] supports tasks spanning up to fifteen minutes by combining dense short-horizon visual context with compressed long-horizon language summaries, and is reused inside $\pi_{0.7}$. Concurrent designs [103], [104], [148] explore different points in the same space, converging on the finding that different time scales need different representations.

World models. Predictive approaches [65], [105], [106], [177] mark a shift from reacting to anticipating, with world-model-generated data reported to improve dual-arm performance by around 30%. The caveat is evaluative rather than

TABLE XIII: Representative real-world deployments of manipulation systems. **No figure in this table is third-party audited**; all reliability and throughput values are self-reported by the operator. "VLA" indicates whether the manipulation policy is a language-conditioned VLA. Non-VLA systems are included because the comparison with deployed scale is informative.

System	Setting	Task	Reported scale and metrics	VLA
Chef Robotics [189], [190]	12+ food facilities	conveyor-synchronized ingredient portioning	10^8 servings; $\sim$2 000 ingredients; -88% giveaway, $+60\%$ labor productivity	-
DYNA-1 [191]	laundries, hospitality	bimanual napkin and towel folding	**99.4% over >24 h, 850+ items, 0 interventions**; $\sim 60\%$ human speed; 98% $\geq 3/5$, 75% at 4 to 5/5	✓
π_0^* [192]	espresso bar, factory, home	espresso, bimanual box assembly, laundry	**18 h continuous**; 59 boxes; 50 novel laundry items; $>90\%$ success; throughput $>2\times$	✓
$\pi_{0.6}$ + partners [200]	warehouses; homes	order packaging; full laundry loads	**175 items/h at 96.4% autonomy over a shift**; interventions/load -50%; missed grasps -42%	✓
Figure 02 (Helix) [188]	automotive body line	sheet-metal loading into a fixture	**1 250+ h**; 90 000+ parts; 30 000+ vehicles; KPI $>99\%$/shift, 0 interventions/shift, 5 mm, 37 s of 84 s takt	✓
Ambi Robotics [193]	national parcel fleet	sortation and stacking	$>250\,000$ **production hours**; 150 M packages; $>1\,200$ parcels/h claimed	-
Amazon Vulcan [194]	fulfillment centers	tactile pick and stow	$>500\,000$ **orders**; covers $\sim 75\%$ of item catalog	-
Agility Digit [195], [196]	third-party logistics	bimanual tote handling	$>100\,000$ **totes**; multi-year service contract	-
BD Stretch [197]	parcel carriers	truck unloading	600 to 800 cases/h; 16 h per charge; **not a VLA**	-
Bright Machines [198]	130+ microfactories	server assembly, human-in-cell	**300 000+ servers**	-
Amazon Blue Jay [199]	fulfillment centers	multi-arm pick, stow, consolidate	**withdrawn** <6 **months after unveiling**; no throughput published	-
$\pi_{0.5}$ [4]	novel real homes	long-horizon cleaning	instruction following 94%, task success 83% (in-distribution)	✓
SRT-H [201]	*ex vivo* surgical	bimanual cholecystectomy sub-steps	**100% on 8 unseen specimens**, no intervention	✓
A-Lab [202]	materials laboratory	synthesis and characterization	**41 novel compounds from 58 targets in 17 days**	-
HarvestFlex [203]	commercial greenhouse	selective harvesting	74.0% success, 32.6 s/pick, 4.1% damage from 3.71 h / 227 episodes	✓

technical: none has been assessed on a shared dual-arm benchmark, so cross-comparison is not currently possible.

Deployment. Section XI adds a dimension the benchmark literature does not capture. Reliability demanded in production, better than 99% placement per shift with zero interventions, inside an 84-second takt [188], exceeds published benchmark rates by a wide margin, and on thirty ordinary real tabletop tasks the best generalist configuration reaches 43.7% [101]. Systems that have accumulated the largest deployed scale remain narrow and are largely not language-conditioned.

Three difficulties have proved durable rather than incidental, and we list them before turning to directions:

- **Contact and force**: position-space policies without force feedback cannot regulate the forces that tightly coupled two-arm tasks require, and difficulty grows with the number of simultaneous contacts.
- **Evaluation**: no shared dual-arm benchmark exists, so cross-method comparison rests on bespoke task suites, and typical trial counts are too small to resolve the differences being claimed.
- **Data**: dual-arm demonstrations remain the most expensive data in robot learning to collect, policies degrade on configurations that data does not cover, and the obvious alternative of learning from autonomous practice is limited by credit assignment over minutes-long sequences.

TABLE XIV: Inference cost and the hardware it implies. "Practical ceiling" is the control rate the reported latency permits for a dual-arm system once chunking is applied, which is the figure that determines whether a policy is deployable rather than merely accurate.

Method	Params	Latency	Hardware / ceiling
RT-2 [1]	55B	$\sim$1 s	TPU class; not real-time
OpenVLA [3]	7B	$\sim$150 ms	A100; $\sim$7 Hz/chunk
RDT-1B [7]	1.2B	$\sim$150 ms	A6000; long H helps
FAST [9]	3B	$\sim$750 ms	RTX 4090; $\sim$1.3 Hz per chunk
π_0 [2]	3B	$\sim$70 ms	A100 or RTX 4090; 50 Hz at H=50
TinyVLA [8]	1B	$\sim$40 ms	RTX 4090; consumer-viable
MiniVLA [88]	$\sim$1B	$\sim$25 ms	RTX 3090; smallest tested
2026 releases publishing per-accelerator figures			
Xiaomi-Robotics-0 [12]	4.7B	$\sim$80 ms	RTX 4090, K=5; 30 Hz
GR00T N1.7 [13]	3B	31 to 173 ms	RTX 5090 to Orin
SmolVLA [95]	450M	$\sim$18 ms	within 1 GB

B. Summary and Discussion

Our principal conclusions, in the order we would defend them:

(1) Coordination comes from generating both arms jointly in chunks, and the reason is geometric rather than expressive. Step-wise prediction cannot express a coordinated two-arm pattern at all, and one second of motion at 50 Hz

happens to match the natural duration of a manipulation phase, which is doing more work than the specific value of H. Producing that chunk for both arms in one pass preserves inter-arm correlation that separate policies would have to reconstruct across an interface. Flow matching earns its place here not by being more expressive than diffusion, which it is not, but because near-straight transport paths permit roughly ten integration steps, and that is what fits an 800-dimensional dual-arm chunk inside a real-time budget; the advantage would evaporate if diffusion sampling became equally cheap. Two parts of this remain inference rather than result. As Table VIII shows, the systems reporting tightly coupled results are concentrated in one family and share no protocol, so the ordering among heads is suggested by the evidence rather than demonstrated by it, and no published work compares a joint against a decomposed policy on the same tightly coupled task under one protocol.

(2) Capability comes from the breadth of what a policy has seen, not the volume of robot data. Policies inheriting web-scale vision-language pre-training consistently beat those trained on robot data alone, because that pre-training supplies object identity, affordance, and spatial language, none of which appears in demonstration trajectories at usable density. The same principle holds within robot data: diversity across environments, objects, and embodiments predicts generalization better than episode count. The caveat from Section VI-E is that this applies to pre-training, since near a task's saturation point distributionally-targeted on-policy episodes outperform additional diverse ones.

(3) Flat imitation caps both reliability and horizon, and there are exactly two demonstrated ways past it. Because dual-arm teleoperation is slow and cautious, imitation inherits a low ceiling, and RECAP-style learning from autonomous experience is the only mechanism shown to exceed it, with an effect size larger than any architectural change of the same period. Separately, flat policies degrade past a few coordinated steps, and separating a reasoning layer from a motor layer converts a long-horizon problem into a sequence of short ones, with the additional benefit that the intermediate language is inspectable. Practice addresses how well a policy does a step; hierarchy addresses how many steps it can chain.

(4) What the field can measure lags what it claims, and what it deploys lags what it demonstrates. Standard suites contain no dual-arm tasks, per-paper task sets are not comparable, and typical trial counts of ten to fifty carry confidence intervals wider than the differences being claimed, so progress is harder to verify than to achieve. Reported latency is subject to the same problem in miniature: overlap scheduling and training-time prefix conditioning have managed the latency-reactivity trade rather than eliminated it, and published figures are means when a control loop must be sized against the tail. The deployment record compounds both. Section XI finds the largest fielded scale in narrow, non-language-conditioned systems, while the most VLA-native industrial deployment measures on the order of a thousand hours. Whether that separation is a lagging indicator or a structural limit is, in our view, the most important open empirical question in the field.

C. Research Directions

(1) A shared bimanual benchmark. Nothing would improve the field's ability to measure itself more cheaply. The requirement is coverage along three axes simultaneously: coupling regime from independent through tightly coupled (Section II-D), object class from rigid through deformable, and horizon from single primitive to multi-minute sequence. The protocol must also fix trial counts large enough to separate the differences papers claim, since at ten to fifty trials the confidence intervals overlap almost everything. This is the one direction on our list blocked by community coordination rather than by research, which is why we place it first: it is the cheapest to act on and the precondition for trusting the rest. Without it, every claim in Section VIII rests on non-comparable evidence.

(2) Dexterity, force, and multi-modal sensing together. Parallel grippers and vision-only observation jointly cap what current systems attempt. Multi-fingered hands would enable in-hand reorientation at the cost of an action space beyond forty dimensions per step; force and tactile channels would supply the contact signal that vision cannot infer; audio marks discrete events such as clicks and seatings. These are one problem rather than three, because each is most useful in the presence of the others: a hand that can reorient an object needs a force signal to know when it is slipping. Simulation is the natural place to acquire the data and also where the difficulty concentrates, since contact and deformation are exactly what current simulators model worst, which makes differentiable contact and randomization targeted at deformable interaction the enabling work.

(3) Safety and reliability arguments that survive deployment. Section XI finds that capability and deployed scale have come apart, and we expect assurance rather than capability to become the binding constraint on commercial two-armed deployment. Systems working near people need more than the rate limits and action clipping surveyed in Section X-B, none of which yields a bound on failure rate; runtime monitoring, constrained generation, and provable avoidance both between the arms and with humans are all required. The reliability a production line demands exceeds published benchmark rates by a wide margin, and that gap will not be closed by a better model alone. Two mechanisms look most promising for closing it, and both depend on the safety argument being available: learning from autonomous experience, which is the only demonstrated route past the ceiling that cautious teleoperation imposes, and few-shot adaptation, since bringing a new two-armed task up from fewer than ten demonstrations would change deployment economics outright.

A further set of directions we regard as important but secondary is discussed where each arises: reactive control above $100\,\mathrm{Hz}$, along with the reporting of latency distributions rather than means (Section VII); composable two-arm primitives under language control (Section IX); collaboration in which one robot arm works with one human arm, which sidesteps the hardest coordination problems by keeping a person in the loop (Section X); and the pairing of persistent memory with predictive world models, where prediction supplies foresight

over seconds and memory supplies state over minutes (Sections VIII and X).

In our assessment, two-armed manipulation has moved in under four years from isolated demonstrations to autonomous operation in unmodified homes, and five factors carried it: capable vision-language backbones, generative action heads that can express multimodal behavior, training recipes that scale and now include learning from experience, affordable dual-arm hardware, and most recently memory and predictive components that stretch the usable horizon. The three directions above describe what separates present systems from reliable ones: the ability to measure progress credibly, sensing that reaches the contact where two arms actually meet, and a safety argument strong enough to deploy behind. Section XI adds a caution that we think should temper the field's reading of its own trajectory: the systems doing the most real work today are not the most capable ones, and closing that gap is at least as much an engineering and evaluation problem as a modeling one.

ACKNOWLEDGMENTS

We thank the authors of the works surveyed here, and the groups who release weights, datasets, and evaluation code, without which a review of this kind would not be possible.

APPENDIX A
GLOSSARY OF ABBREVIATIONS

Abbrev.	Expansion
Models and policy classes	
VLA	vision-language-action model
VLM	vision-language model
AR	autoregressive (action generation)
DiT	diffusion transformer
FM	flow matching
Learning settings	
BC	behavioral cloning
IL	imitation learning
RL	reinforcement learning
RECAP	RL from autonomous capability
Execution and representation	
BID	bidirectional decoding
RTC	real-time chunking
TTAC	training-time action conditioning
KI	knowledge insulation
Auxiliary components	
MEM	multi-scale embodied memory
WM	world model
DVA	direct video action
General	
DOF	degrees of freedom
ODE	ordinary differential equation
OXE	Open X-Embodiment

REFERENCES

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "RT-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

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

[3] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, M. Nasiriany, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "OpenVLA: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[4] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, "$\pi_{0.5}$: A vision-language-action model with open-world generalization," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2025.

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gober, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalber, S. Levine, Y. Lin, H. Luu, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "RT-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[6] A. Amin, K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, and H. Zhang, "$\pi^*_{0.6}$: A vla that learns from experience," *arXiv preprint arXiv:2511.14759*, 2025.

[7] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "RDT-1B: A diffusion foundation model for bimanual manipulation," *arXiv preprint arXiv:2410.07864*, 2024.

[8] J. Wen, Y. Zhu, J. Zhang, M. Mu, Z. Qi, Z. Peng, G. Wan, T. Li, J. Huang, and H. Lu, "TinyVLA: Towards fast, data-efficient vision-language-action models for robotic manipulation," *arXiv preprint arXiv:2409.12514*, 2024.

[9] K. Pertsch, M. J. Kim, J. Luo, S. Levine, and C. Finn, "FAST: Efficient action tokenization for vision-language-action models," in *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

[10] B. Ai, A. Amin, R. Aniceto, A. Balakrishna, K. Black, D. Driess, C. Finn, K. Hausman, B. Ichter, S. Levine, K. Pertsch, L. X. Shi, J. T. Springenberg, Q. Vuong, M. Torne, A. Z. Ren, S. Nair, K. Stachowicz, A. Walling, U. Zhilinsky *et al.*, "$\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities," *arXiv preprint arXiv:2604.15483*, 2026.

[11] Gemini Robotics Team, Google DeepMind, "Gemini robotics on-device 2: Model card," Google DeepMind Model Cards, July 2026, accessed: 2026-07-31. [Online]. Available: https://deepmind.google/models/model-cards/gemini-robotics-on-device-2/

[12] R. Cai, J. Guo, X. He, P. Jin, J. Li, B. Lin, F. Liu, W. Liu, F. Ma, K. Ma, F. Qiu, H. Qu, Y. Su, Q. Sun, D. Wang, D. Wang, Y. Wang, R. Wu, D. Xiang, Y. Yang, H. Ye, Y. Zhang, and Q. Zhou, "Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution," *arXiv preprint arXiv:2602.12684*, 2026.

[13] NVIDIA, "NVIDIA Isaac GR00T N1.7: Open reasoning vision-language-action model for humanoid robotics," NVIDIA / Hugging Face technical report, 2026, https://huggingface.co/blog/nvidia/gr00t-n1-7.

[14] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman, B. Ichter, D. Driess, J. Wu, C. Lu, and M. Schwager, "Foundation models in robotics: Applications, challenges, and the future," *The International Journal of Robotics Research*, vol. 43, no. 14, pp. 2164–2204, 2024.

[15] R. Wolf, Y. Shi, S. Liu, and R. Rayyes, "Diffusion models for robotic manipulation: A survey," *Frontiers in Robotics and AI*, vol. 12, 2025.

[16] M. Abbas, J. Narayan, and S. K. Dwivedy, "A systematic review on cooperative dual-arm manipulators: Modeling, planning, control, and vision strategies," *International Journal of Intelligent Robotics and Applications*, vol. 7, pp. 683–707, 2023.

[17] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation—a survey," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1340–1353, 2012.

[18] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision (IJCV)*, vol. 128, pp. 261–318, 2020.

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

[20] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[21] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, and M. Nickel, "Flow matching for generative modeling," *arXiv preprint arXiv:2210.02747*, 2022.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[26] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[27] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[28] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[29] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, "Flamingo: A visual language model for few-shot learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[30] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[31] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[32] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[33] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[34] Gemini Team, "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[35] Google DeepMind, "Gemini 2.0 flash thinking: Experimental model," *Technical Report*, 2025.

[36] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "PaLM-E: An embodied multimodal language model," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

[37] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, T. Unterthiner, D. Keysers, A. Arnab, and X. Zhai, "PaLIGemma: A versatile 3b vlm for transfer," *arXiv preprint arXiv:2407.07726*, 2024.

[38] Gemma Team, "Gemma: Open models based on Gemini research and technology," *arXiv preprint arXiv:2403.08295*, 2024.

[39] L. X. Shi, B. Ichter, M. Equi, S. Levine, and K. Hausman, "Hi robot: Open-ended instruction following with hierarchical vision-language-action models," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

[40] D. A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems (NeurIPS)*, 1989.

[41] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[42] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[43] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive networks: Self-supervised learning from video," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[44] S. Ross, G. J. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[45] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[46] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[47] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[48] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[49] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[50] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[51] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[52] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2022.

[53] N. M. M. Shafiullah, Z. Cui, A. Altanzaya, and L. Pinto, "Behavior transformers: Cloning k modes with one stone," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[54] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[55] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Giménez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas, "A generalist agent," in *Transactions on Machine Learning Research (TMLR)*, 2022.

[56] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research (IJRR)*, 2023.

[57] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," *arXiv preprint arXiv:2209.03003*, 2022.

[58] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorber, A. Sauer, F. Boesel, D. Podell, T. Dockhorn, Z. English, K. Lacey, A. Goodwin, Y. Marek, and R. Rombach, "Scaling rectified flow transformers for high-resolution image synthesis," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[59] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile ALOHA: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2024.

[60] Open X-Embodiment Collaboration, "Open X-Embodiment: Robotic learning datasets and RT-X models," *arXiv preprint arXiv:2310.08864*, 2024.

[61] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, Y. You, A. Khazatsky, S. Karamcheti, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[62] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, M. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. Florence, K. Fang, D. Sadigh, S. Levine, and C. Finn, "DROID: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[63] H. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Florence, and S. Levine, "BridgeData V2: A dataset for robot learning at scale," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[64] F. Ebert, Y. Yang, K. Schmeckpeper, B. Buber, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, "Bridge data: Boosting generalization of robotic skills with cross-domain datasets," in *Proceedings of Robotics: Science and Systems (RSS)*, 2021.

[65] GigaAI, "GigaBrain-0.5M*: A vla with world model-based reinforcement learning," *arXiv preprint arXiv:2602.12099*, 2025.

[66] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "LIBERO: Benchmarking knowledge transfer for lifelong robot learning," *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[67] X. Li, K. Hsu, J. Liu, K. Pertsch, Q. Vuong, and S. Levine, "SIMPLER: Simulated manipulation policy evaluation for real robot setups," *arXiv preprint*, 2024.

[68] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "RLBench: The robot learning benchmark and learning environment," in *IEEE Robotics and Automation Letters (RA-L)*, 2020.

[69] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," *Proceedings of the Conference on Robot Learning (CoRL)*, 2020.

[70] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," in *arXiv preprint arXiv:2009.12293*, 2020.

[71] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, X. Yuan, P. Xie, Z. Huang, R. Chen, and H. Su, "ManiSkill2: A unified benchmark for generalizable manipulation skills," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[72] C. Lee, R. Zhang, J. Zhu, F. Xia, K. Ehsani, R. Martín-Martín, Y. Li, S. Savarese, and L. Fei-Fei, "BEHAVIOR-1K: A human-centered, embodied AI benchmark with 1,000 everyday activities and realistic simulation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2024.

[73] X. Li, K. Hsu, J. Liu, K. Pertsch, Q. Vuong, and S. Levine, "Evaluating real-world robot manipulation policies in simulation," *arXiv preprint*, 2024.

[74] Open X-Embodiment Collaboration, "Open x-embodiment: Robotic learning datasets and RT-X models," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[75] K. Wu, X. Wu, R. Zhang, J. Duan, B. Ni, J. Lu, J. Zheng, and H. Fan, "GR-1: Unleashing large-scale video generative pre-training for visual robot manipulation," in *arXiv preprint*, 2023.

[76] Y. Li, K. Fang, K. Hausman, B. Ichter, and P. Florence, "Hamster: Hierarchical action models for open-world robot manipulation," *arXiv preprint*, 2025.

[77] D. Chen, J. Wang, Y. Xu, R. Zhang, X. Li, Y. Wu, J. Shao, and L. Ke, "SpatialVLA: Exploring spatial representations for visual-language-action model," *arXiv preprint*, 2024.

[78] S. Haldar, J. Pari, A. Rao, M. Watter, and L. Pinto, "BAKU: An efficient transformer for multi-task policy learning," in *arXiv preprint*, 2024.

[79] M. J. Kim, K. Pertsch, D. Sadigh, S. Levine, and C. Finn, "KAT: Keypoint-action tokens for robot manipulation," *arXiv preprint*, 2024.

[80] H. Zhen, Y. Qiu, and S. Sun, "SimpleVLA-RL: Simple vision-language-action models meet reinforcement learning," *arXiv preprint*, 2024.

[81] D. Driess, J. T. Springenberg, B. Ichter, L. Yu, A. Li-Bell, K. Pertsch, A. Z. Ren, H. Walke, Q. Vuong, L. X. Shi, and S. Levine, "Knowledge insulating vision-language-action models: Train fast, run fast, generalize better," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[82] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Fang, J. Jiang, H. Fan, and N. Duan, "CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation," *arXiv preprint*, 2025.

[83] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[84] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox, "RVT: Robotic view transformer for 3d object manipulation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[85] Y. Ze, G. Yan, Y. Wu, Y. Jia, R. Zhang, Y. Hu, J. Wu, J. Tan, H. Sun, H. Su, and X. Wang, "3D diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[86] C. Zhou, L. Yu, A. Babu, K. Tirumala, M. Yasunaga, L. Shamber, J. Kahn, X. Ma, L. Zettlemoyer, and O. Levy, "Transfusion: Predict the next token and diffuse images with one multi-modal model," *arXiv preprint arXiv:2408.11039*, 2024.

[87] J. Chen, S. Li, K. Zhang, Y. Chen, R. Ding, Y. Ge, and J. Zhao, "HybridVLA: Collaborative diffusion and autoregression in a unified vision-language-action model," *arXiv preprint arXiv:2503.10631*, 2025.

[88] S. Belkhale and D. Sadigh, "MiniVLA: A better vla with a smaller footprint," *arXiv preprint*, 2024.

[89] Gemini Robotics Team, "Gemini robotics: Bringing AI into the physical world," *arXiv preprint arXiv:2503.20020*, 2025, google DeepMind technical report. [Online]. Available: https://arxiv.org/abs/2503.20020

[90] ——, "Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer," *arXiv preprint arXiv:2510.03342*, 2025, google DeepMind technical report; v3 revised 28 Nov 2025. [Online]. Available: https://arxiv.org/abs/2510.03342

[91] Google DeepMind, "Gemini robotics on-device model card," Google DeepMind Model Cards, June 2025, announced 24 June 2025. Based on on-device Gemma models. Accessed: 2026-07-31. [Online]. Available: https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-Robotics-On-Device-Model-Card.pdf

[92] Gemini Robotics Team, Google DeepMind, "Gemini robotics 2 brings whole body intelligence to robots," Google DeepMind Blog, July 2026, accessed: 2026-07-31. [Online]. Available: https://deepmind.google/blog/gemini-robotics-2-brings-whole-body-intelligence-to-robots/

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

[94] S. Liu, B. Li, K. Ma, L. Wu, H. Tan, X. Ouyang, H. Su, and J. Zhu, "RDT2: Exploring the scaling limit of UMI data towards zero-shot cross-embodiment generalization," *arXiv preprint arXiv:2602.03310*, 2026.

[95] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans, S. Palma, A. Zouitine, M. Aractingi, C. Pascal, M. Russi, A. Marafioti, S. Alibert, M. Cord, T. Wolf, and R. Cadene, "SmolVLA: A vision-language-action model for affordable and efficient robotics," *arXiv preprint arXiv:2506.01844*, 2025.

[96] Figure AI, "Helix: A vision-language-action model for generalist humanoid control," Figure AI technical blog, February 2025, https://www.figure.ai/news/helix, accessed 2026-07-31.

[97] ——, "Introducing helix 02: Full-body autonomy," Figure AI Technical Blog, Jan. 2026, https://www.figure.ai/news/helix-02, accessed 2026-07-31.

[98] C. Cheang, S. Chen, Z. Cui, Y. Hu, L. Huang, T. Kong, H. Li, Y. Li, Y. Liu, X. Ma, H. Niu, W. Ou, W. Peng, Z. Ren, H. Shi, J. Tian, H. Wu, X. Xiao, Y. Xiao, J. Xu, and Y. Yang, "GR-3 technical report," *arXiv preprint arXiv:2507.15493*, 2025.

[99] 1X Technologies, "Redwood ai: An end-to-end model for whole-body mobile manipulation on neo," 1X Technologies Technical Blog, Jun. 2025, https://www.1x.tech/discover/redwood-ai, accessed 2026-07-31.

[100] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," in *Proceedings of Robotics: Science and Systems (RSS)*, 2025, arXiv:2502.19645.

[101] A. Yakefu, B. Xie, C. Xu, E. Zhang, E. Zhou, F. Jia, H. Yang, H. Fan, H. Zhang, H. Peng, J. Tan, J. Huang, K. Liu, K. Liu, K. Gu, Q. Zhang, R. Zhang, S. Huang, S. Cheng, S. Liu, T. Wang, T. Wang, W. Sun, W. Tang, Y. Wei, Y. Chen, Y. Gui, Y. Zhao, Y. Ma, Y. Wei, Y. Yang, Y. Guo, Z. Chen, Z. Du, Z. Zhang, Z. Liu, and Z. Yan, "RoboChallenge: Large-scale real-robot evaluation of embodied policies," *arXiv preprint arXiv:2510.17950*, 2025.

[102] M. Torne, K. Pertsch, H. Walke, K. Vedder, S. Nair, B. Ichter, A. Z. Ren, H. Wang, J. Tang, K. Stachowicz, K. Dhabalia, M. Equi, Q. Vuong, J. T. Springenberg, S. Levine, C. Finn, and D. Driess, "MEM: Multi-scale embodied memory for vision language action models," *arXiv preprint arXiv:2603.03596*, 2026.

[103] H. Shi, X. Bin, Y. Liu, L. Sun, L. Fengrong, T. Wang, E. Zhou, H. Fan, X. Zhang, and G. Huang, "MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation," *arXiv preprint arXiv:2508.19236*, 2025.

[104] H. Jang, S. Yu, H. Kwon, H. Jeon, Y. Seo, and J. Shin, "ContextVLA: Vision-language-action model with amortized multi-frame context," *arXiv preprint arXiv:2510.04246*, 2025.

[105] GigaAI Team, "GigaBrain-0: A world model-powered vision-language-action model," *arXiv preprint arXiv:2510.19430*, 2025.

[106] Rhoda AI, "Causal video models are data-efficient robot policy learners," https://www.rhoda.ai/research/direct-video-action, 2026.

[107] WorldVLA Team, "WorldVLA: Towards autoregressive action world model," *arXiv preprint arXiv:2506.21539*, 2025.

[108] GigaAI Team, "GigaWorld-0: World models as data engine to empower embodied ai," *arXiv preprint arXiv:2511.19861*, 2025.

[109] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[110] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[111] M. Xiao, Z. Song, N. Sebe, and L. Cheng, "Align-then-steer: Aligning vision-language-action models for efficient robot policy fine-tuning," *arXiv preprint*, 2025.

[112] J. Yang, M. Torne, S. Bahl, A. R. M. B. Villalonga, and S. Levine, "Robot fine-tuning made easy: Pre-training rewards and policies for autonomous real-world reinforcement learning," *arXiv preprint*, 2025.

[113] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2022.

[114] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar, "RoboAgent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[115] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2018.

[116] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-Learning for offline reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[117] X. B. Peng, A. Kumar, G. Zhang, and S. Levine, "Advantage-weighted regression: Simple and scalable off-policy reinforcement learning," in *arXiv preprint arXiv:1910.00177*, 2019.

[118] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[119] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," in *arXiv preprint arXiv:1707.06347*, 2017.

[120] R. Ranawaka Aracchige, C. Chi, S. Song, and B. Burchfiel, "SAIL: Sample-efficient policy adaptation for faster-than-demonstration execution," *arXiv preprint*, 2025.

[121] J. Lu, J. Luo, K. Pertsch, and S. Levine, "VLA-RL: Reinforcement learning for vision-language-action models," *arXiv preprint*, 2025.

[122] J. Chen, Y. Yuan, J. Li, and Y. Qiao, "ConRFT: A reinforced fine-tuning method for vla models via consistency regularization," *arXiv preprint*, 2025.

[123] Y. Chebotar, Q. Vuong, K. Hausman, F. Xia, Y. Lu, A. Irpan, A. Kumar, T. Yu, A. Herzog, K. Pertsch, K. Gopalakrishnan, J. Ibarz, O. Vinyals, and S. Levine, "Q-Transformer: Scalable offline reinforcement learning via autoregressive q-functions," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[124] A. Z. Ren, J. Dinh, S. Dai, T. Zhang, M. Phielipp, and B. Burchfiel, "Diffusion policy policy optimization," *arXiv preprint*, 2023.

[125] S. K. S. Ghasemipour, P. Florence, S. Lazzaro, and S. Levine, "Self-improving foundation models for embodied intelligence," *arXiv preprint*, 2025.

[126] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research (IJRR)*, vol. 37, no. 4–5, pp. 421–436, 2018.

[127] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[128] J. Zheng, J. Yao, D. Yu, Z. Zhao, T. Wang, L. Pan, L. Zhang, Y. Yang, and J. Zou, "Universal actions for enhanced embodied foundation models," *arXiv preprint*, 2025.

[129] L. Ke, K. Pertsch, S. Levine, and C. Finn, "Consistency models as a rich and efficient policy class for reinforcement learning," in *arXiv preprint*, 2024.

[130] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang, "GNFactor: Multi-task real robot learning with generalizable neural feature fields," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2023, arXiv:2308.16891.

[131] Y. Dai, S. Bahl, A. Singh, K. Pertsch, and S. Levine, "RACER: Rich language-guided failure recovery policies for imitation learning," *arXiv preprint*, 2024.

[132] K. Black, K. Pertsch, S. Nair, and S. Levine, "Real-time chunking: Real-time execution of action chunking flow policies," *arXiv preprint*, 2025.

[133] Y. Liu, J. I. Hamid, A. Xie, Y. Lee, M. Du, and C. Finn, "Bidirectional decoding: Improving action chunking via guided test-time sampling," *arXiv preprint arXiv:2408.17355*, 2024.

[134] K. Black, K. Pertsch, S. Nair, and S. Levine, "Training-time action conditioning for efficient real-time chunking," *arXiv preprint*, 2025.

[135] J. Grannen, Y. Chong, T. Z. Zhao, and C. Finn, "Stabilize to act: Learning to coordinate for bimanual manipulation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[136] R. Chitnis, S. Tulsiani, S. Gupta, and A. Gupta, "Efficient bimanual manipulation using learned task schemas," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[137] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[138] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gober, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do as I can, not as I say: Grounding language in robotic affordances," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2022.

[139] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, "Mimicplay: Long-horizon imitation learning by watching human play," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[140] L. Chen, S. Bahl, and D. Pathak, "Playfusion: Skill acquisition via diffusion from language-annotated play," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[141] Z. Cui, H. Wang, N. M. M. Shafiullah, and L. Pinto, "From play to policy: Conditional behavior generation from uncurated robot data," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[142] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tompson, D. Schuurmans, and P. Abbeel, "Learning universal policies via text-guided video generation," *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[143] Y. Hu, F. Xie, W. Jia, G. Wang, J. Zhao, and Y. Gao, "Look before you leap: Unveiling the power of GPT-4V in robotic vision-language planning," in *arXiv preprint*, 2023.

[144] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman, "LIV: Language-image representations and rewards for robotic control," in *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023, arXiv:2306.00958.

[145] H. S. Li, Y. Yang, X. Chen, X. Chen, Y. Yang, H. Tian, T. Wang, D. Lin, and F. Zhao, "CronusVLA: Towards efficient and robust manipulation via multi-frame vision-language-action modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[146] M. S. Mark, J. Liang, M. Attarian, C. Fu, D. Dwibedi, D. Shah, and A. Kumar, "BPP: Long-context robot imitation learning by focusing on key history frames," *arXiv preprint arXiv:2602.15010*, 2026.

[147] M. Torne, A. Tang, Y. Liu, and C. Finn, "Learning long-context diffusion policies via past-token prediction," *arXiv preprint arXiv:2505.09561*, 2025.

[148] H. Fang, M. Grotz, W. Pumacay, Y. R. Wang, D. Fox, R. Krishna, and J. Duan, "SAM2Act: Integrating visual foundation model with a memory architecture for robotic manipulation," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

[149] A. Sridhar, J. Pan, S. Sharma, and C. Finn, "MemER: Scaling up memory for robot control via experience retrieval," *arXiv preprint arXiv:2510.20328*, 2025.

[150] Y.-L. Wei, H. Liao, Y. Lin, P. Wang, Z. Liang, G. Liu, and W.-S. Zheng, "CycleManip: Enabling cyclic task manipulation via effective historical perception and understanding," *arXiv preprint arXiv:2512.01022*, 2025.

[151] C. Chi, Z. Xu, and S. Song, "UMI on legs: Making manipulation policies mobile with locomotion," *arXiv preprint*, 2025.

[152] R. Shah, R. Martín-Martín, and Y. Zhu, "BUMBLE: Unifying reasoning and acting with vision-language models for building-wide mobile manipulation," *arXiv preprint*, 2024.

[153] AgiBot Team, "AgiBot World: A new frontier for generalist robot policies," *arXiv preprint*, 2025.

[154] Physical Intelligence, "Scaling robot policy learning via zero-shot labeling with foundation models," *arXiv preprint*, 2025.

[155] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "BC-Z: Zero-shot task generalization with robotic imitation learning," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2022.

[156] O. Mees, L. Hermann, and W. Burgard, "What matters in language conditioned robotic imitation learning over unstructured data," in *IEEE Robotics and Automation Letters (RA-L)*, 2022.

[157] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

[158] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "LLM-Planner: Few-shot grounded planning for embodied agents with large language models," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

[159] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "VoxPoser: Composable 3d value maps for robotic manipulation with language models," *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[160] S. Wang, M. Li, G. Liang, and M. Qiao, "LLM^3: Large language model-based task and motion planning with motion failure reasoning," *arXiv preprint*, 2024.

[161] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Manipulate-anything: Automating real-world robots using vision-language models," *arXiv preprint*, 2024.

[162] J. Duan, S. Nasiriany, H. Li, and A. Mandlekar, "Manipulate-Anything: Automating real-world robots using vision-language models," *arXiv preprint*, 2024.

[163] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, B. Zitkovich, F. Xia, C. Finn, and K. Hausman, "Open-world object manipulation using pre-trained vision-language models," *arXiv preprint*, 2023.

[164] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, "Interactive language: Talking to robots in real time," in *IEEE Robotics and Automation Letters (RA-L)*, 2023.

[165] Z. Xian, N. Gkanatsios, T. Gerber, T.-W. Ke, and K. Fragkiadaki, "Chain-of-thought predictive control," in *arXiv preprint*, 2023.

[166] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine, "Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2024, arXiv:2408.11812.

[167] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian *et al.*, "RT-H: Action hierarchies using language," in *arXiv preprint*, 2024.

[168] P. Liu, Y. Orru, C. Paxton, N. M. M. Shafiullah, and L. Pinto, "OK-Robot: What really matters in integrating open-knowledge models for robotics," *arXiv preprint*, 2024.

[169] H. Etukuru, S. Nair, J. Pari, A. Padmanabha, G. Kamat, S. Dasari, T. Arbuckle, B. Maddukuri, A. Juliani, and S. Levine, "Robot utility models: General policies for zero-shot deployment in new environments," *arXiv preprint*, 2024.

[170] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2021.

[171] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3M: A universal visual representation for robot manipulation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2022.

[172] S. Nair, E. Mitchell, K. Chen, S. Savarese, C. Finn, and D. Sadigh, "Learning language-conditioned robot behavior from offline data and crowd-sourced annotation," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2022.

[173] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berber, P. Mathur, T. Olkin, D. Raber, D. Srinivasan, J. Hong, and D. Batra, "Where are we in the search for an artificial visual cortex for embodied intelligence?" in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[174] H. Cheng, C. Cheang, Y. Li, J. Yang, and H. Lu, "SPA: 3d spatial-awareness enables effective embodied representation," *arXiv preprint*, 2024.

[175] H. Bharadhwaj, R. Mottaghi, S. Tulsiani, and A. Gupta, "Gen2Act: Human video generation in novel scenarios enables generalizable robot manipulation," in *arXiv preprint*, 2024.

[176] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani, "Track2Act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation," in *arXiv preprint*, 2024.

[177] Y. Wu *et al.*, "Video prediction policy: A generalist robot policy with predictive visual representations," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

[178] ViPRA Team, "ViPRA: Video prediction for robot actions," *arXiv preprint arXiv:2511.07732*, 2025.

[179] Mimic-Video Team, "Mimic-video: Video-action models for generalizable robot control beyond VLAs," *arXiv preprint arXiv:2512.15692*, 2025.

[180] FOFPred Team, "Future optical flow prediction improves robot control and video generation," *arXiv preprint arXiv:2601.10781*, 2026.

[181] Meta AI, "V-JEPA 2: Self-supervised video models enable understanding, prediction and planning," *arXiv preprint arXiv:2506.09985*, 2025.

[182] UP-VLA Team, "UP-VLA: A unified understanding and prediction model for embodied agent," *arXiv preprint arXiv:2501.18867*, 2025.

[183] NVIDIA, "Cosmos: World foundation models for physical AI," *arXiv preprint*, 2025.

[184] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu, Q. Vuong, T. Zhang, T.-W. Lee, K.-H. Lee, P. Xu, S. Kirmani, Y. Zhu, B. Ichter, J. Tompson, S. Levine, and F. Xia, "PIVOT: Iterative visual prompting elicits actionable knowledge for vlms," *arXiv preprint*, 2024.

[185] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "TidyBot: Personalized robot assistance with large language models," in *Autonomous Robots*, 2023.

[186] I. R. T. Xiao, L. Pinto, and J. Malik, "Robot learning with sensorimotor pre-training," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[187] Z. Liu, C. Chi, E. Cousineau, N. Kuppuswamy, B. Burchfiel, and S. Song, "ManiWAV: Learning robot manipulation from in-the-wild audio-visual data," *arXiv preprint*, 2025.

[188] Figure AI, "F.02 contributed to the production of 30,000 cars at bmw," Figure AI Blog, Nov. 2025, https://www.figure.ai/news/production-at-bmw, accessed 2026-07-31.

[189] Chef Robotics, "Chefos: Physical ai for food manufacturing," Chef Robotics product documentation, 2026, nSF/ANSI 169 certified robot module C-001748. [Online]. Available: https://www.chefrobotics.ai/

[190] ——, "Chef robotics completes 100 million product servings milestone," Reported by The Robot Report, Apr. 2026, https://www.therobotreport.com/chef-robotics-completes-100-million-product-servings-milestone/, accessed 2026-07-31.

[191] Dyna Robotics, "DYNA-1: The first commercial-ready robot foundation model offering fully autonomous, round-the-clock dexterity," Dyna Robotics press release, Apr. 2025, reports 99.4% success without human intervention over $>$24 h continuous autonomous operation at 60% of human throughput; https://www.dyna.co/.

[192] P. Intelligence, A. Amin, R. Aniceto, A. Balakrishna, K. Black, K. Conley, G. Connors, J. Darpinian, K. Dhabalia, J. DiCarlo, D. Driess, M. Equi, A. Esmail, Y. Fang, C. Finn, C. Glossop, T. Godden, I. Goryachev, L. Groom, H. Hancock, K. Hausman, G. Hussein, B. Ichter, S. Jakubczak, R. Jen, T. Jones, B. Katz, L. Ke, C. Kuchi, M. Lamb, D. LeBlanc, S. Levine, A. Li-Bell, Y. Lu, V. Mano, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, C. Sharma, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, W. Stoeckle, A. Swerdlow, J. Tanner, M. Torne, Q. Vuong, A. Walling, H. Wang, B. Williams, S. Yoo, L. Yu, U. Zhilinsky, and Z. Zhou, "$\pi^*_{0.6}$: a VLA that learns from experience," *arXiv preprint arXiv:2511.14759*, 2025.

[193] Ambi Robotics, "Ambios and ambisort: Ai-powered parcel sortation," Ambi Robotics product documentation, 2026. [Online]. Available: https://www.ambirobotics.com/

[194] Amazon, "Introducing Vulcan: Amazon's first robot with a sense of touch," About Amazon, May 2025, picks and stows $\approx$75% of warehouse items; $>$500,000 orders processed in Spokane, WA and Hamburg, Germany pilots; https://www.aboutamazon.com/news/operations/amazon-vulcan-robot-pick-stow-touch.

[195] Agility Robotics, "Digit moves over 100,000 totes in commercial deployment," Agility Robotics Blog, Nov. 2025, https://www.agilityrobotics.com/content/digit-moves-over-100k-totes, accessed 2026-07-31.

[196] GXO Logistics, "GXO signs industry-first multi-year agreement with agility robotics," GXO Newsroom, Jun. 2024, first formal commercial humanoid RaaS contract; https://gxo.com/news_article/gxo-signs-industry-first-multi-year-agreement-with-agility-robotics/.

[197] Boston Dynamics, "Dhl group signs mou for additional 1,000-robot deployment," Boston Dynamics newsroom, 2025. [Online]. Available: https://bostondynamics.com/news/dhl-signs-mou-for-additional-1000-robot-deployment/

[198] Bright Machines, "Bright machines expands bright factory platform with hybrid BRC for resilient AI infrastructure production," Bright Machines press release, Jul. 2026, 130+ microfactories, 60+ customers, 300,000+ servers produced; https://www.globenewswire.com/news-release/2026/07/29/3335279/0/en/Bright-Machines-Expands-Bright-Factory-Platform-With-Hybrid-BRC-for-Resilient-AI-Infrastructure-Production.html.

[199] Amazon, "Introducing Blue Jay and Project Eluna, amazon's latest robotics and AI technology for its operations," About Amazon, Oct. 2025, project discontinued February 2026, less than six months after announcement; https://www.aboutamazon.com/news/operations/new-robots-amazon-fulfillment-agentic-ai.

[200] Physical Intelligence, "Real-world deployment performance of $\pi_{0.6}$ with weave robotics and ultra," Physical Intelligence technical update, Feb. 2026, reports 50% reduction in teleoperator interventions per laundry load and $>$80% autonomy in e-commerce packing. Reported by Humanoids Daily, https://www.humanoidsdaily.com/news/the-api-fication-of-robotics-physical-intelligence-unveils-real-world-performance-data-with-weave-and-ultra.

[201] J. W. Kim, J.-T. Chen, P. Hansen, L. X. Shi, A. Goldenberg, S. Schmidgall, P. M. Scheikl, A. Deguet, B. M. White, D. R. Tsai, R. Cha, J. Jopling, C. Finn, and A. Krieger, "SRT-H: A hierarchical framework for autonomous surgery via language conditioned imitation learning," *arXiv preprint arXiv:2505.10251*, 2025.

[202] N. J. Szymanski, B. Rendy, Y. Fei, R. E. Kumar, T. He, D. Milsted, M. J. McDermott, M. Gallant, E. D. Cubuk, A. Merchant, H. Kim, A. Jain, C. J. Bartel, K. Persson, Y. Zeng, and G. Ceder, "An autonomous laboratory for the accelerated synthesis of novel materials," *Nature*, vol. 624, no. 7990, pp. 86–91, 2023.

[203] Z. Zhao, S. Wang, Z. Miao, and Y. Xiong, "Harvestflex: Strawberry harvesting via vision-language-action policy adaptation in the wild," *arXiv preprint arXiv:2603.05982*, 2026.

[204] R. Sala Sisó, T. Silvério, J. Sand, and T. N. Le, "Closing the lab-to-store gap: A data-efficient post-training and experience-driven learning vla framework for retail humanoids," *arXiv preprint arXiv:2607.20345*, 2026.