Scholar Publishing Group
International Journal of Neural Network
https://doi.org/10.38007/NN.2026.050115
ISSN 2790-2692 Vol. 5, Issue 1: 142-155

# *Application of Multimodal Large Models Combining Vision and Language In Intelligent Robots*

**Qingyao Jia**

*New York University, Tandon School of Engineering, Brooklyn, 11201, New York, United States*

***Keywords:*** Vision-language-action model; Intelligent robot; Embodied intelligence; Evidence synthesis; Runtime assurance; Robot manipulation

***Abstract:*** Vision-Language-Action (VLA) foundation models extend robot learning from task-specific policies to open-vocabulary, language-conditioned manipulation, but published benchmark results do not independently validate a new deployment architecture. This evidence-synthesis paper reviews representative VLA systems and datasets from 2023-2026, distinguishes between peer-reviewed evidence, preprints and official technical sources, and re-presents public OpenVLA results with their original evaluation scope. In the published BridgeData V2 WidowX suite, the mean success rates are 18.5% for RT-1-X, 20.0% for Octo, 50.6% for RT-2-X, and 70.6% for OpenVLA; in the Franka-Tabletop adaptation study, pretrained OpenVLA fine-tuned on target data reaches a trial-weighted 67.2% over 99 rollouts. These figures support claims about the cited models only. Based on the above evidence, this paper proposes a semantic slow-loop, action fast-loop and independent runtime-assurance layer, along with embodiment-aware training, edge/local execution roles, staged validation and a non-compensatory evaluation dashboard. The framework is shown as a conceptual engineering guideline for implementation and formal safety analysis of controlled robot validation before the deployment claim can be made.

## 1 Introduction

Intelligent robotics does not have an inherent conflict between "understanding" and "action". Traditional modular systems can perform precise, verifiable and safety-constrained motion in structured environments by exposing intermediate states that support feasibility checks, fault isolation, component replacement and certification through perception, task planning, trajectory generation and servo control. The main deficiency occurs when the category of objects, instructions, viewpoints or environments deviates from the assumptions embedded in the modules. Vision-language foundation models offer an alternative path by aligning images, text and robot states in a common representation space to enhance open-vocabulary grounding and task transfer. PaLM-E incorporates continuous sensor representations into a language model and demonstrates embodied multimodal reasoning and robot planning [1], but it should be viewed as evidence for high-level embodied reasoning rather than high-frequency motor control.

RT-2 further maps robot actions into tokens and co-fine-tunes web-scale vision-language data

Copyright: © 2026 by the authors. This is an Open Access article distributed under the Creative Commons Attribution License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (https://creativecommons.org/licenses/by/4.0/).

with robot trajectories, transferring semantic knowledge to end-to-end control and demonstrating generalization to new objects, instructions and semantic relationships in large-scale real-robot evaluation [2]. Open X-Embodiment subsequently built a cross-embodiment data initiative and RT-X policy families by harmonizing data from 22 robot embodiments and over a million trajectories [3]. The two are not the same: the dataset initiative and the RT-X models/evaluations are separate entities, and the evidence supports cross-embodiment transfer rather than a single universal architecture.

RoboCat shows that a self-improvement cycle of policy execution, additional trajectory collection and retraining can support adaptation to new robot arms and skills with relatively little target-task data [4], although it is not a language-conditioned VLA. Together, these studies have shown a shift from semantic planning to action generation and adaptation, and have also revealed a core engineering trade-off: open-world semantic flexibility must be linked with kinematic feasibility, contact dynamics, low-latency feedback and independently enforceable safety constraints. Therefore, the public benchmark gains cannot be considered proof that any new fusion, loss, safety controller or edge-cloud architecture has been validated.

Therefore, this paper will take the form of an evidence synthesis and conceptual deployment-framework study rather than an experimentally validated methods paper. It addresses the following four questions: (1) How do recent VLA systems differ in terms of data, action representation, embodiment coverage, adaptation and validation? (2) What can and cannot be concluded from public OpenVLA benchmark aggregates? (3) How can semantic planning, continuous action generation, low-level control and final safety authority be separated through explicit interfaces? (4) Which deployment and evaluation practices are required before real-world claims are justified? The contributions are a source-aware taxonomy, a bounded secondary analysis of public results, a proposed three-loop/five-module architecture with formal runtime-assurance semantics, and a staged governance and evaluation protocol. The rest will present the evidence basis, identify the scope and deployment difficulties of the evidence, introduce the proposed framework, and finally mention limitations and validation requirements.

Review Protocol and Evidence Classes. Based on the above evidence synthesis, the primary sources available up to July 2026 have been organised around the search concepts "vision-language-action robot", "generalist robot policy", "cross-embodiment", "robot manipulation benchmark", "VLA safety", "confidence calibration", and "runtime assurance". Peer-reviewed conference/journal publications are considered the strongest evidence for published technical and numerical claims; primary preprints are used to describe recent systems whose publication status has been explicitly stated; and official model cards or technical documentation are only employed for deployment characteristics that are not presented as peer-reviewed results. Inclusion refers to direct relevance for robot manipulation, embodied reasoning, VLA evaluation, adaptation, safety or deployment. Public benchmark values are kept with their source version, denominator, scoring rule and published uncertainty (if available), and conceptual recommendations are labelled as proposals rather than empirical results.

## 2 Evidence Synthesis and Taxonomy of VLA Systems

### 2.1 The Data Infrastructure has switched from platform-specific collection to cross-embodiment sharing.

Foundation-policy performance is severely limited by data coverage. Robot trajectories do not occur naturally in the internet-scale text and image corpora; they require physical hardware, teleoperation, synchronisation and safety supervision. Platform differences in coordinate systems, control frequencies, gripper definitions and sensors also reduce reuse. DROID has a single

collection hardware and protocol for all environments, and has reported over 76,000 demonstration trajectories and 350 hours of interaction across 564 scenes [5]. This supports the extension from high consistency in a single testbed to long-tail coverage across buildings, operators and tasks, but the published scale should be referred to as scene coverage rather than as proof of cross-platform robustness.

Open X-Embodiment and DROID both put forward data governance plans; the former coordinates heterogeneous robot embodiments and action spaces, and the latter extends environmental diversity under a unified collection system. Therefore, engineering datasets need more than trajectory counts. Record synchronous sensor timestamps, coordinate transformations, language annotations, failed and interrupted trajectories, intervention events, and versioned calibration metadata. Industrial environments may also need tool coordinates, load and force/torque signals; home-service environments require unstructured placement, human interference and open-vocabulary commands. These are deployment prerequisites, not proof that a VLA model will automatically generalize after aggregation.

### 2.2 Model architectures have moved from semantic planning to language-conditioned action generation.

Octo is an open-source generalist robot policy pretrained on roughly 800,000 trajectories and subsequently fine-tuned for downstream embodiments; its Transformer policy and diffusion-style action head support language or goal-image conditioning and rapid adaptation across multiple robot platforms [6]. Continuous-action policies can represent multimodal motion distributions and smooth trajectories; autoregressive action tokenization can reuse the language-model training machinery but at the cost of longer decoding sequences. The above ways should be viewed as different action interfaces rather than as a single progression in which one representation is universally superior.

OpenVLA combines DINOv2 and SigLIP visual features with a Llama-2-based language backbone, is trained on about 970,000 real-robot demonstrations, and its peer-reviewed CoRL/PMLR publication reports good multi-robot results and practical fine-tuning mechanisms [7]. π0, which has now been formally published at RSS 2025, connects a vision-language backbone with a flow-matching action expert for continuous robot control [8]. The flow-matching process does not automatically show a calibrated action entropy that can be interpreted as physical risk. Model selection should therefore account for action representation, embodiment coverage, adaptation cost, latency and the type of validation actually reported.

*Table 1: Evidence-Oriented Taxonomy of Representative Vision-Language-Action Systems*

| System / entity type | Year / evidence | Architecture and action representation | Embodiment / data coverage | Inputs and adaptation | Validation / safety status |
|---|---|---|---|---|---|
| PaLM-E / model | 2023 / peer-reviewed | Embodied multimodal language model; high-level reasoning | Multi-domain embodied data | Image, text, robot state | Embodied reasoning/planning; not high-frequency motor-control evidence |
| RT-2 / VLA policy | 2023 / peer-reviewed | Autoregressive VLA; tokenized robot actions | Robot trajectories + web VLM data | Image, language; web/robot co-fine-tuning | Real-robot semantic-to-action transfer; no independent hard-safety guarantee |

| System / entity type | Year / evidence | Architecture and action representation | Embodiment / data coverage | Inputs and adaptation | Validation / safety status |
|---|---|---|---|---|---|
| Open X-Embodiment / dataset initiative + RT-X policy family | 2024 / peer-reviewed | Cross-embodiment dataset and Transformer policy families | 1M+ trajectories; 22 embodiments | Image, language; heterogeneous embodiment transfer | Cross-robot positive transfer; not one unambiguous architecture |
| Octo / generalist policy | 2024 / peer-reviewed | Transformer + diffusion-style action head | ~800k pretraining trajectories | Language or goal image; downstream fine-tuning | Pretrained then fine-tuned; task-dependent performance |
| OpenVLA / open-source VLA | 2025 / peer-reviewed | 7B VLM + discrete action tokens | ~970k demonstrations | Image, language; target-domain fine-tuning/LoRA | Published multi-robot evaluation; safety layer not validated by benchmark |
| π0 / VLA flow model | 2025 / peer-reviewed | VLM + flow-matching action expert | Cross-embodiment dexterous data | Image, language; continuous action generation | High-frequency action modeling; action entropy is not calibrated physical risk |
| Gemini Robotics VLA / Gemini Robotics-ER | 2025 / preprint + official family | VLA control separated from embodied reasoning capabilities | Multi-embodiment robot data | Vision, language, action; adaptation to new settings | Broad demonstrations; evidentiary status differs from peer-reviewed deployment validation |

### 2.3 Published BridgeData V2 evidence supports model-level comparisons but not causal attribution.

This paper re-plots, rather than independently verifies, the BridgeData V2 WidowX aggregate evaluation reported by OpenVLA [7] and grounded in the BridgeData V2 dataset [9]. The suite has 17 task items and 10 rollouts per task, totaling 170 rollouts for the model; selected tasks allow for partial credit. The means and error bars reproduced below are published aggregates and published standard errors. No independent pairwise significance tests or multiple-comparison corrections are carried out in this paper.

Data sources: Kim et al. [7], Table 4; BridgeData V2 context [9]. Error bars are published standard errors, not independent estimates.

As shown in Figure 1, the reported mean success rates are 18.5% for RT-1-X, 20.0% for Octo, 50.6% for RT-2-X and 70.6% for OpenVLA. OpenVLA is therefore significantly higher in this reported evaluation, but this difference cannot be solely attributed to internet vision-language pretraining, as the compared systems also differ in robot-data scale, cleaning, visual encoders, architecture and adaptation. The figure shows a descriptive model-level comparison of the WidowX suite and does not show that a particular component of the proposed framework causes the gain.

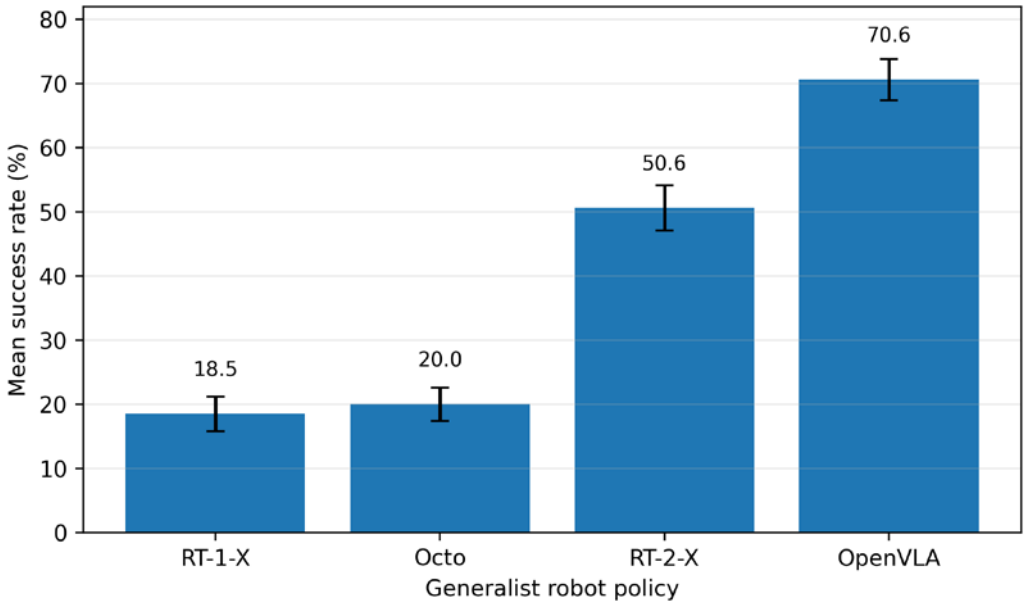



*Figure 1: Replot of Published Mean Success Rates on the BridgeData V2 WidowX Evaluation*

### 2.4 The scope of application is expanding from single-step operation to long-horizon, cross-embodiment applications.

OpenVLA-OFT reports an increase in the average success rate of the four LIBERO tasks from 76.5% to 97.1% and a 26-fold improvement in action-generation throughput through parallel decoding, action chunks and an optimised continuous-action fine-tuning design [10]. Gemini Robotics' VLA control model is different from Gemini Robotics-ER embodied reasoning, and it has demonstrated open-vocabulary physical interaction in multiple robot environments [11]. Recently, overlapping systems have also reduced the novelty of a generic slow/fast decomposition: GR00T N1 uses a dual-system VLA with a vision-language module and diffusion-transformer action module [12]; π0.5 combines heterogeneous co-training with semantic subtask prediction and low-level actions for long-horizon open-world manipulation [13]; SmolVLA aims for efficient and asynchronous inference on affordable hardware [14]; and Gemini Robotics On-Device reports local execution and low-latency adaptation in the official technical documentation [15]. The latter three sources include preprint or official-technical evidence and are used here as recent design context, not as peer-reviewed validation of this paper's proposed deployment framework.

## 3 Evidence Boundaries and Deployment Challenges

### 3.1 The semantic-to-action chain requires explicit interfaces for grounding and physical feedback.

Vision-language models can suggest what to do and identify probable objects, but a robot still needs to realise an executable chain: instruction → grounding → affordance → target pose → kinematic feasibility → trajectory → contact/force control → feedback. The command "put the cup in the sink" can be realised in various ways according to the shape of the cup, the position of grasping, camera angle, obstacle arrangement and arm reach. Action tokenization provides a convenient language model interface but discretises pose and time; continuous heads have richer motion structure but require embodiment-aware control and feedback. Force-related commands will

only work if force/torque sensing, suitable labels or an explicit force controller are available. Semantic correctness is therefore not dynamic feasibility or a safety guarantee.

### 3.2 Generalisation failures can be divided into perceptual, semantic, physical, embodiment and temporal shifts.

Cross-scene generalization is not a single failure mode. Perceptual shift includes occlusion, transparency, reflections and lighting; semantic shift covers new object names, relations and ambiguous references; geometric/physical shift includes size, friction, deformability and contact conditions; embodiment shift covers camera extrinsics, kinematics, grippers and control interfaces; and temporal/human-interaction shift includes moved objects, sudden intervention and long-horizon error accumulation. Internet image-text knowledge can improve recognition under some semantic shifts, but it lacks the robot-centric depth, contact and action consequences required for physical transfer. Therefore, the assessment will report the performance of each shift type and not employ a single average failure rate.

### 3.3 Model size, communication delay and control-loop timing are all relatively small.

Real robots generally have multiple timing layers instead of a single control frequency. High-level semantic planning may be event-driven or low-rate; learned action inference operates at a moderate rate; and low-level servo and safety monitoring can run at hundreds of hertz or higher depending on the hardware. Large VLA backbones cannot be assumed to meet these budgets at the edge without measurement. Fully cloud-based execution introduces network jitter, stale state, privacy and availability risks, and fully local execution has memory, power and thermal constraints. Action chunks reduce repeated decoding but increase open-loop exposure if the horizon is too long. Any deployment claim therefore requires measured p50/p95 latency, bandwidth, stale-state behavior, failover and disturbance-sensitive horizon tuning.

### 3.4 Physical Safety, Functional Safety, Cybersecurity, Privacy and Accountability all require independent controls.

Open-vocabulary instructions can be ambiguous, malicious or inconsistent with the observed scene. A model may ground a non-existent target, misinterpret a reference, or remain confidently wrong under distribution shift. These cases should not be classified as a single "security" problem. Physical and functional safety require explicit limits, safe-state logic and independent final authority; cybersecurity demands authentication, isolation and command-integrity controls; privacy needs data minimisation and access governance; and malicious-command risk requires policy checks and refusal. Clarification, re-observation, assistance requests, refusal, controller override and emergency stop are all different interventions. Model confidence can inform escalation but cannot be equated with a calibrated probability of physical safety or used as a substitute for legal responsibility.

## 4 Proposed Safety-Aware Deployment Framework and Validation Protocol

### 4.1 Proposed Hierarchical Vision-Language-Action Architecture with Explicit Loop Ownership

The framework below is a conceptual design synthesis and not an implemented system. The five functional modules are mapped to three runtime loops: a semantic slow loop (language planning

and semantic grounding), an action fast loop (perception/fusion and short-horizon action generation), and an independent safety loop with final command authority. The slow loop produces a structured goal z_t containing target object, spatial relation, requested action and termination condition. The fast loop receives multi-view visual history, depth/geometry and proprioceptive state, and then generates embodiment-specific target poses, gripper commands or force references. The safety loop receives the proposed command and current robot/safety sensor data, either projects it into the admissible set, overrides it with a fallback controller, or triggers a hard stop. Nominal update-rate ranges in Table 2 are design placeholders that must be measured and tuned for a specific robot.

$$h_t = LN(W_s \varphi_s(I^{M}_{t-T:t}) + W_g \varphi_g(D^{M}_{t-T:t}) + W_l \varphi_l(q,z_t) + W_p \varphi_p(s_t)) \quad (1)$$

As shown in Equation (1), $I_{t-T:t}^{1:M}$ is a temporal window of M camera views, $D_{t-T:t}^{1:M}$ is depth or geometric features, q is the user command, $z_t$ is the structured subgoal/termination state, and $s_t$ is the proprioceptive robot state. $\varphi_s$ and $\varphi_g$ are separate semantic and geometric visual encoders; $\varphi_l$ and $\varphi_p$ encode language/subgoals and proprioception; $W_*$ project all streams to a common d-dimensional space before layer normalization. This explicitly implements the dual visual branches mentioned in the architecture and defines the state passed to the action policy. Reachability and workspace checks are not assumed to emerge from $h_t$; they are still explicit downstream constraints.

Before action generation, the object, geometric and temporal tokens can be aligned with the structured subgoal using standard scaled dot-product attention. Let $Q_t$ be queries derived from $z_t$, $K_t$ and $V_t$ be keys and values from the visual/temporal tokens, and $M_t$ be a mask encoding missing views, invalid depth or disallowed token associations. The generated context is:

$$c_t = \mathrm{softmax}((Q_t K_t^T / \sqrt{d}) + M_t)V_t \quad (2)$$

Equation (2) is the standard scaled attention form applied to explicitly constructed tokens, and it is not considered an explanation or calibrated confidence score. Grounding supervision should link language spans or structured goal fields to object/region tokens, and occlusion evaluation should test multi-view consistency, re-observation, and failure to ground. For commands such as "grab the red bottle instead of the eggplant", the attention can support attribute-location binding, but the policy still needs to pass geometric feasibility and safety checks before execution.

*Table 2: Proposed Modules, Runtime Loops and Failure-Handling Interfaces*

| Module / runtime loop | Input / state owner | Core operation / nominal update mode | Output / interface | Failure handling / final authority |
|---|---|---|---|---|
| Perception / action fast loop | Multi-view RGB-D and recent frames | Semantic + geometric encoding; refresh on new observation | Object, geometry and temporal tokens | Invalidate cached features after camera/object motion, lighting change or tracking loss |
| Language planner / semantic slow loop | User instruction, scene context, prior subtask | Structured goal parsing; event-driven or low-rate | $z_t$: target, relation, action, termination | Clarify, re-observe or refuse when grounding/intent is ambiguous |
| Cross-modal fusion / slow-to-fast bridge | Visual tokens, $z_t$, proprioceptive state | Masked scaled attention and state alignment | Unified latent state $h_t$ | Calibration/OOD signals may request re-observation or assistance; not hard safety |

| Module / runtime loop | Input / state owner | Core operation / nominal update mode | Output / interface | Failure handling / final authority |
|---|---|---|---|---|
| Action policy / action fast loop | h_t and robot embodiment | Receding-horizon action chunks; illustrative 5–20 Hz | Pose, gripper and optional force references | Tune horizon K under disturbances; fall back to servo/force controller when needed |
| Safety filter / independent safety loop | Proposed command, robot state, safety sensors | Constraint projection/override; platform-dependent 100 Hz–1 kHz | Admissible command or fallback/stop | Final control authority; hard limits, safe set, watchdog and emergency stop |

Table 2 addresses the problem of the three loops and five modules by assigning a runtime owner to each module and setting up an explicit message interface. The semantic loop can request targets and subtasks, but it cannot bypass the feasibility or safety checks directly. The action loop generates candidate motion commands for the current embodiment. The independent safety filter has the final say and must be implementable without relying on the generative model's self-assessment. This decomposition retains the transfer benefits of VLA reasoning and keeps intermediate verification, fault isolation and deterministic fallback paths.

## 4.2 Proposed Embodiment-Aware Training and Adaptation Objective

The training objective of this conceptual framework should separate heterogeneous command components and not apply a single unnormalized L1 loss to position, rotation, gripper and force. For an action chunk, normalised robust regression can be used for translation; a rotation-appropriate geodesic or quaternion loss should be employed for orientation; classification/regression is suitable for gripper state; and subtask/termination labels can supervise the structured goal interface. Cross-modal alignment is still auxiliary, and explicit failure, intervention and recovery labels provide targets for safety-related prediction. A compact form is:

$$L = \lambda_p L_{pos} + \lambda_R L_{rot} + \lambda_g L_{grip} + \lambda_s L_{subtask} + \lambda_c L_{align} + \lambda_r L_{safe/recovery} \quad (3)$$

Equation (3) is used to compute L_pos after embodiment-specific normalisation; L_rot is a rotation-aware distance in SO(3) or an equivalent normalised quaternion loss; L_grip handles discrete or continuous gripper states; L_subtask is cross-entropy over structured subgoals/termination; L_align is a contrastive or cosine alignment term; and L_safe/recovery supervises intervention, pre-collision, recovery or stop labels. The loss weights should be reported with units/normalisation. During adaptation, the trainable set should be specified explicitly (visual backbone, language backbone, adapters, alignment layers, action head and embodiment embeddings) because broader visual or low-rank adaptation may be required under scene shift.

Organise the training data into general successful trajectories, physically valid robustness augmentation, and failure/recovery supervision. Language rewriting is acceptable only when task meaning is preserved; viewpoint perturbation, background replacement and object rearrangement must preserve camera geometry, depth, contact and trajectory consistency or regenerate the affected supervision. Failure data should distinguish between unsafe approach, collision/near-collision, force-limit event, human intervention, recovery and benign task failure. Action-chunk horizon K should be tuned under sudden obstacles, contact changes and observation delay rather than fixed a priori. Therefore, augmentation, freezing and chunk length are used as validation parameters instead of fixed values.

### 4.3 Formal Runtime Assurance replaces entropy-based command blending

Action-distribution entropy is not a normalised measure of physical risk and may be unavailable or intractable for deterministic, diffusion or flow-matching policies. A convex combination of the two commands does not generally preserve actuator, collision or safe-set constraints. Therefore, the proposed deployment architecture will replace the previous entropy gate with a formal runtime-assurance filter. Let u_t^VLA be the candidate command and h_j(x) ≥ 0 define explicit safe-set constraints. A control-barrier-function style projection [16] can be written as:

$$u_t^* = \arg\min_{u \in U} ||u - u_t^{VLA}||_{R^2} \text{ s.t. } \dot{h}_j(x_t,u) + \alpha_j h_j(x_t) \geq 0, \forall j \quad (4)$$

Equation (4) projects the candidate command into an admissible control set U that satisfies explicit safety constraints; if the optimisation is infeasible, stale, delayed or triggered by a hard hazard, the runtime-assurance layer must reject the VLA command and switch to a certified fallback or emergency stop. Therefore, the final authority does not need model confidence. Recent VLA safety-alignment work, such as SafeVLA [17] and confidence-calibration studies [18], has addressed when to request help or refrain from acting, but these signals are not yet linked to physical constraint enforcement. Evaluation should report false-negative safety events, time-to-intervention, recovery success and any constraint violations, and not assume that a low-entropy action is safe.

### 4.4 Real-time deployment requires a limited time budget, local safety authorities and explicit edge/cloud failure behaviour.

A defensible deployment split keeps cloud functions mostly offline: large-scale pre-training, data cleaning, fleet analytics and versioned model preparation. An edge workstation can host the vision-language backbone and event-driven semantic planning, and the robot controller will execute embodiment-specific action interfaces, low-level servoing, and an independent safety filter locally. Illustrative Design ranges are 0.2-2 Hz for semantic replanning, 5-20 Hz for learned action updates, and 100 Hz-1 kHz for safety/servo functions; however, these are not measured results and need to be replaced by platform-specific timing budgets. The system should set a worst-case latency, watchdog timeout, authentication, bandwidth limit, failover mode and behaviour when observations or model state are stale.

Computation reuse is permitted only after explicit cache invalidation. Static-background features should be invalidated after camera motion, tracked-object displacement, significant lighting changes or tracking confidence loss; otherwise, the system needs to re-encode the observation. Quantisation and low-rank adaptation can reduce compute and adaptation costs, but the alignment and action-output paths need to be task-specifically sensitive-tested. Knowledge distillation can be considered for target selection or subtask decomposition, yet no compression scheme is assumed safe by default. Deployment reports should include p50/p95 end-to-end latency, action-update jitter, energy or power, intervention rate, safety-filter activation, stale-state events and recovery performance.

### 4.5 Public-evidence reanalysis and a non-compensatory multi-criteria evaluation framework

OpenVLA's published Franka-Tabletop adaptation experiments include single-instruction, multi-object, multi-instruction and distribution-shift scenarios. The following figure re-plots the published aggregate values from the CoRL/PMLR source version. The bars are model conditions from the source and not experiments conducted in this paper; the error bars are published standard errors.

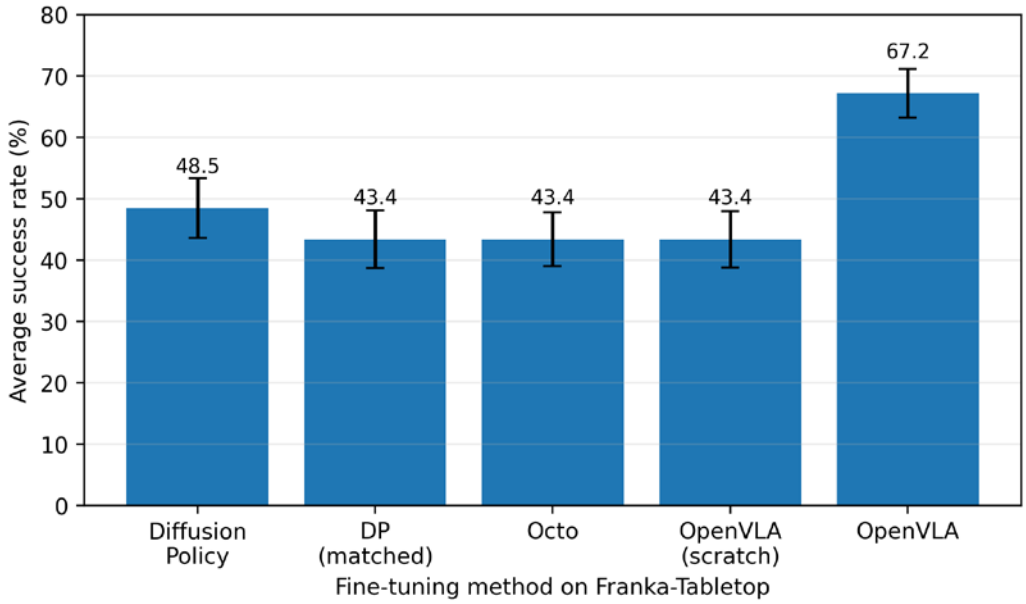



*Figure 2: Replot of Published Franka-Tabletop Downstream Adaptation Results*

Data source: Kim et al. [7], Table 6 in the proceedings/early edition. Error bars are published standard errors; later arXiv versions may renumber the table.

As shown in Figure 2, Diffusion Policy [19] has a 48.5% trial-weighted aggregate; the matched-data Diffusion Policy condition, Octo and scratch-trained OpenVLA each show 43.4%; and pretrained OpenVLA after Franka-Tabletop target-data fine-tuning reaches 67.2% over 99 rollouts. Octo should be referred to as a pre-trained generalist policy that is then fine-tuned, not as a scratch-trained condition. Matched-data baseline helps to separate data effects from architecture effects. The 67.2% figure is specific to Franka-Tabletop adaptation and not a general "new robot" success rate; the same OpenVLA source reports an average of 58.3% for Franka-DROID [7]. No single method is suitable for all problems; therefore, task-stratified results are more informative than a single aggregate.

A single weighted score may overlook unacceptable safety or latency by compensating for it with task success. Deployment evaluation should therefore use a dashboard with confidence intervals, hard acceptance thresholds and Pareto analysis. Long-horizon benchmarks, such as VLABench [20], and the new unified evaluation infrastructure vla-eval [21] have shown that grounding, spatial reasoning, embodiment transfer, execution length and reproducibility need to be separated. A condensed metric vector is:

$$M = [S_{task}, G_{shift}, L_{p50/p95}, E, V_{safe}, I_{human}, R_{recovery}, C_{cal}]^T \quad (5)$$

In Equation (5), S_task is the task success reported by task category; G_shift is generalization through perceptual, semantic, physical, embodiment and temporal shifts; L_p50/p95 reports latency distribution; E records energy or compute cost; V_safe counts or rates physical constraint violations; I_human records interventions with reason codes; R_recovery measures recovery success and time; and C_cal reports calibration/selective-risk behaviour where probabilistic confidence exists. Safety and functional limits are the acceptance constraints, not conditions to be traded for success. Confidence intervals and per-task denominators should be shown with means, and Pareto fronts are used when efficiency, latency and success conflict.

*Table 3: Proposed Deployment Decision Matrix with Evidence and Acceptance Boundaries*

| Scenario / hazard assumption | Main sensors | Proposed control stack | Evidence status | Example acceptance criteria | Standards / residual risk |
|---|---|---|---|---|---|
| Industrial assembly / rigid contact, tight tolerances | RGB-D, force/torque, proprioception | VLA task/pose proposal + force/visual servo + independent safety filter | Conceptual; requires cell-specific validation | No hard-limit violation in acceptance set; force and p95 latency below predefined limits | ISO 10218-1/-2, ISO 13849-1; residual collision/tolerance risk |
| Warehouse picking / occlusion, moving assets | Multi-view RGB-D, barcode; AMR sensing if mobile | Action-chunk VLA + visual servo; AMR navigation separated when applicable | Conceptual; benchmark evidence is not site validation | Task-stratified pick rate; zero protected-zone violations; stale-state watchdog tested | ISO 10218; ISO 3691-4 when driverless mobile platform is used |
| Home service / close human proximity, open vocabulary | RGB-D, audio, tactile sensing | Hierarchical VLA + clarification/refusal + local safety filter | Conceptual; needs human-proximity and ambiguity testing | Predefined stop distance; intervention and false-grounding thresholds; recovery test | ISO 13482 where applicable; privacy and unpredictable-human-motion risk |
| Mobile manipulation / navigation-manipulation coupling | LiDAR, RGB-D, odometry, proprioception | Task planner + certified navigation safety + VLA manipulation interface | Conceptual; long-horizon validation required | Route and manipulation success reported separately; failover and stale-map tests passed | ISO 3691-4 where applicable; accumulated planning/localization error |
| Human-robot collaboration / shared workspace | Vision, force/torque, proximity or wearable signals | Shared control + force/impedance control + independent runtime assurance | Conceptual; physical acceptance testing mandatory | Speed/separation or force limits verified; emergency-stop and recovery timing tested | ISO 10218 and ISO/TS 15066; unexpected human motion remains a key hazard |

Table 3 is a proposed decision matrix and not evidence that the framework has already achieved success in the listed areas. Connect each application to hazard assumptions, sensor prerequisites, independent low-level control, acceptance criteria and a standards context. The matrix also avoids circular recommendations; a VLA is not "recommended" just because the proposed framework contains one. Each deployment needs to show that its perception, action interface, safety function, timing budget and recovery behaviour meet the scenario-specific acceptance criteria before limited release.

### 4.6 Staged Implementation, Shadow-Mode Limits and Continuous-Learning Governance

A proposed implementation life cycle includes task definition, data audit, offline training,

simulation, isolated physical testing, shadow evaluation, limited rollout and versioned production release. Natural-language tasks should first be translated into observable success, failure and safety conditions. Simulation can show many failure modes, but it cannot replace controlled contact tests. Shadow mode can measure disagreement, latency, predicted constraint violations and calibration, but it cannot show the actual results of actions that have not been taken. Therefore, the promotion will be a limited-robot trial with an upper acceptance limit and independent logging/rollback. Only after passing through the above gates will the operating range be extended.

Continuous learning in production should still be managed, not self-directed. New on-site data should be versioned with model, robot, sensor-calibration and software identifiers; sensitive recordings require access and retention controls. Candidate updates are trained offline and evaluated using regression suites for previous tasks, known shift categories, latency, safety-filter behaviour and recovery. Drift detection should prompt a re-assessment, not automatic promotion. Release authority, freeze criteria, approval responsibility and rollback triggers must be named in the deployment process, and emergency safety functions should remain outside the self-updating learning loop.

## 5 Limitations and Conclusion

Vision-Language-Action models offer a convenient interface for open-vocabulary perception, language-conditioned goals and robot action generation, but the strongest published evidence is still model- and benchmark-specific. The re-plotted public OpenVLA results show significant aggregate advantages in the cited BridgeData V2 WidowX and Franka-Tabletop evaluations, but also reveal confounds such as data scale, cleaning, visual encoders, architecture and adaptation. They do not validate the multimodal fusion equation, proposed training objective, runtime-assurance filter, edge/local deployment split or decision matrix introduced in this paper.

Therefore, the contribution of this study is a limited evidence synthesis and a safety-aware conceptual deployment framework. It separates semantic slow-loop reasoning, action fast-loop control and an independent safety authority; replaces entropy-based command blending with explicit constraint projection/override; treats confidence as an escalation signal rather than physical risk; uses embodiment-aware training targets and physically consistent augmentation; and evaluates deployment with non-compensatory metrics, acceptance thresholds and staged governance. The main deficiencies are the lack of original implementation and robot experiments. To support a validated-method claim, a formal safety analysis is required, along with controlled physical experiments, VLA and continuous-action baselines, component ablations, latency and energy measurements, calibration and failure-case analysis, reproducible configurations and logs. Until the above conditions are met, the framework will be used as a reference for the following implementation and evaluation.

## References

[1] *Driess D, Xia F, Sajjadi MSM, et al. PaLM-E: An Embodied Multimodal Language Model. Proceedings of the 40th International Conference on Machine Learning, PMLR, 2023, 202: 8469-8488.*

[2] *Zitkovich B, Yu T, Xu S, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. Proceedings of the 7th Conference on Robot Learning, PMLR, 2023, 229: 2165-2183.*

[3] *Zhang, Y. (2026). Exploration of Enterprise Big Data Microservice Architecture Based on Domain-Driven Design (DDD). Procedia Computer Science, 282, 1994-2003.*

[4] *Zhang, Z. (2026). Research on Service-oriented Packaging of AI Systems and Data-driven Decision Support for Real-world Business Processes. Engineering Advances, 6(3).*
[5] *Zhang, Y. (2025, January). Research on the Application and Optimization of Multi Dimensional Data Model Based on Kylin in Enterprise Technology Management. In International Conference on Innovative Computing (pp. 234-241). Singapore: Springer Nature Singapore.*
[6] *Jingzhi Yin. (2026). Constructing a CMBS Default Risk Sentiment Index Using Financial Text Embeddings and Evaluating Its Predictive Effectiveness. Engineering Advances, 6(1), 50-54.*
[7] *Fei, W., Xue, W., & Wang, B. (2025). Research on the innovative path of rural education teaching methods from the perspective of PBE and PBL integration. Int. Acad. J. Humanit. Soc. Sci.*
[8] *Fei, W. (2026). Families Of Chinese Background In Rural Alabama: Education, Immigration, And Identity.*
[9] *Wendan Xue, Wen Fei. An Analytical Study on the Differences and Determinants of Basic Education Resource Allocation in Rural China and the United States. International Journal of Educational Research and Development, 1(3), 13-17.*
[10] *Boli Wang, Wen Fei. Innovations and practices in the classroom for rural aesthetic education in the perspective of school integration. International Journal of Educational Research and Development, 3(1), 36-41.*
[11] *Fei, W., & Qian, J. (2026). Application and Exploration of Reverse Instructional Design in the Teaching of the Primary School Physical Education" Walking and Playing" Unit. Journal of Sociology and Education, 2(2).*
[12] *Wen Fei, Jianrong Qian. Research on Value Guidance and Practical Pathways for Constructing the Professional Culture of Teachers. International Journal of Educational Research and Development, 2(4), 27-30.*
[13] *Qian, X. (2026). Intelligent Supply Chain Collaboration Models in Retail Supported by Omnichannel Marketing Data. Advances in Computer and Communication, 7(3).*
[14] *Su, J. (2026). Android Communication Reliability Enhancement Architecture Based on Multi Link Diversity Transmission.*
[15] *Liu, S. (2026). Design and Implementation of Retrieval Enhancement Generation Method for Enterprise Knowledge Base.*
[16] *Su, J. (2026, July). Research on Machine Learning-based Abnormal Traffic Detection and Anti-Interference Methods for Android End-to-End Encrypted Transmission in High-Reliability Application Scenarios. In 2026 5th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) (pp. 1-6). IEEE.*
[17] *Qian, X. (2026, July). Application of the Combination Model of Prophet and LSTM in Multi-Level Inventory Demand Prediction for Retail Supply Chain of CPG-Consumer Packaged Foods. In 2026 5th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) (pp. 1-7). IEEE.*
[18] *Jin, C. (2026). Construction and Empirical Study of Credit Risk Assessment Index System for Small and Micro Enterprises in Commercial Banks.*
[19] *Jin, C. (2026). Application of Data Mining Technology in Predicting User Churn and Retention Strategies in E-Commerce.*
[20] *Zhang, Z. (2026). User Behavior Anomaly Identification Integrating Telemetry Data and Its Application in Consumer Finance Risk Control.*
[21] *Liu, S. (2026). Automatic Generation Method and Implementation of Web Front-End Components Based on Generative AI.*

*[22] Xin, H. (2026). Research on Distributed Data Pipelines and Event-Driven Architecture for Freight Visibility.*

*[23] Zhang, Z. (2026). User Behavior Device Reliability Joint Modeling and Intelligent Risk Control Practice In Consumer Technology.*

*[24] Wang, B. (2025, November). Research on Performance Improvement of Personalized Recommendation System for E-commerce Platform Based on Big Data and Collaborative Filtering Algorithm. In International Conference on Computational Technologies for Research in Data Analytics (pp. 36-46). Cham: Springer Nature Switzerland.*

*[25] Wang, Z. (2026). Research on Operation Data Analysis and Efficiency Improvement of the New Energy Industry Based on Machine Learning. Procedia Computer Science, 285, 605-612.*

*[26] Ma, W. (2026). Research on Reliability Enhancement of Cloud Data Platform Driven by Adaptive Sharding and Fault Tolerance Algorithms. Procedia Computer Science, 285, 1020-1029.*

*[27] Chi, C. (2026, July). Credit Risk Dynamic Pricing Representation Learning Algorithm based on Long-Term SelfSupervised Learning. In 2026 5th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) (pp. 1-7). IEEE.*

*[28] Zhang, Y. (2026). A Framework for Semantic Alignment and Privacy Compliance of Large-Scale Language Models for Cross-Border Data Governance. Procedia Computer Science, 285, 1039-1047.*

*[29] Sun, L. (2026, July). Low-Latency Optimized Architecture for Distributed Stream Processing Systems for RealTime Bidding Advertising. In 2026 5th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) (pp. 1-7). IEEE.*

*[30] Sun, X. (2026). Research on Online Identification and Alarm Mechanism for Abnormal Power Quality in Data Centers.*