# Bridging the Pretrain-to-Real Gap: Alignment Challenges in Deploying Generalist VLA Models for Additive Manufacturing

Zhugang Liu[1]
zhugang.liu01@utrgv.edu

Kaichuang Zhang[2]
kaichuangzhang@usf.edu

Qi Lu[1]
qi.lu@utrgv.edu

Efren Saenz[1]
efren.saenz01@utrgv.edu

Martha Asare[1]
martha.asare01@utrgv.edu

Maxim Ermolinsky[3]
maxim.ermolinsky01@utrgv.edu

Jose Hernandez[3]
jose.hernandez112@utrgv.edu

Jinghao Yang[3,*]
jinghao.yang@utrgv.edu

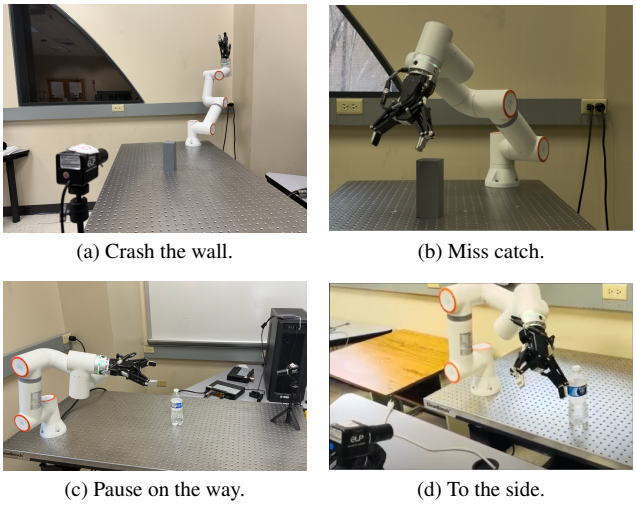
[1]Department of Computer Science, The University of Texas Rio Grande Valley
[2]Department of Electrical Engineering, University of South Florida
[3]Department of Electrical and Computer Engineering, The University of Texas Rio Grande Valley

## Abstract

*Generalist Vision-Language-Action (VLA) models mark a significant milestone in the Generative AI era. However, the prevailing reliance on simulated benchmarks obscures the severe physical and algorithmic domain shifts encountered during real-world hardware deployment. This paper presents a comprehensive, high-performance framework that successfully deploys the 7B-parameter OpenVLA model directly onto a physical Franka Research 3 (FR3) robotic arm. Utilizing a meticulously optimized distributed cloud-edge architecture, we specifically target the highly dynamic workflows of additive manufacturing. Differentiating from simulation-only studies, we identify and decisively resolve the "system-level alignment challenges" intrinsic to this Pretrain-to-Real transition. Using the grasping of topologically complex 3D-printed parts as a rigorous evaluation metric, we detail our algorithmic mitigation strategies—including deterministic decoding enforcement, strict thresholding from continuous to binary, and geometric retargeting combined with Low-Rank Adaptation (LoRA). Our empirical results demonstrate that the aligned hardware system achieves a 98% reduction in spatial L2 error and ensures stable convergence of gripper states, providing a robust, field-tested taxonomy for deploying generative embodied AI in flexible industrial settings.*

## 1. Introduction

Recent advances in Generative AI have accelerated the development of Vision-Language-Action (VLA) models, which aim to unify visual perception, language understanding, and low-level action prediction within a single policy. Large language models have substantially expanded the reasoning and instruction-following capacity of embodied systems [5, 29], while powerful visual encoders have improved visual representation quality for downstream robotic perception and grounding [20, 23, 31]. Building on these advances, recent VLA systems such as RT-2 and OpenVLA demonstrate that natural-language instructions can



(a) Crash the wall. (b) Miss catch. (c) Pause on the way. (d) To the side.

Figure 1. While zero-shot VLA baselines fail on complex out-of-distribution additive manufacturing products due to severe spatial mirroring, our proposed physical alignment pipeline effectively resolves these shifts and enables robust terminal targeting in real-world deployment.

be grounded into embodied manipulation behaviors, opening a promising path toward more general-purpose robotic autonomy[4, 13]. In particular, OpenVLA provides an open-weight 7B-scale VLA model built upon large-scale real-world robotic demonstrations, making it a practical foundation for downstream adaptation and deployment in real robotic systems [13, 19].

This trend is especially relevant to additive manufacturing environments, where production increasingly follows a High-Mix, Low-Volume (HMLV) pattern. In such settings, the product mix, part geometry, and manipulation requirements may vary substantially across jobs, creating significant challenges for fixed, manually engineered automation pipelines [2, 14]. Traditional robotic manipulation pipelines often rely on manually designed kinematic waypoints, accurate geometric models, or task-specific grasp templates, which can make adaptation more difficult when object geometry and workspace conditions vary significantly [27]. By contrast, VLA models offer a more flexible alternative: instead of solving each new object family with a separately engineered pipeline, a single embodied policy can, in principle, interpret language instructions and adapt its manipulation behavior from visual observations.

Despite this promise, most existing progress in generalist robot learning has been validated either on standardized benchmarks, highly curated research environments, or cloud-dependent pipelines [3, 4, 30]. For industrial post-processing, however, practical deployment requires more than benchmark success. Real manufacturing systems impose strict constraints on data privacy, system latency, hardware reliability, and contact-rich execution, all of which are difficult to capture faithfully in simulation alone [11, 16, 32]. As a result, a substantial gap remains between the strong pretraining-time capability of modern VLAs and their robust execution on local physical hardware.

This gap becomes particularly evident when open-weight VLAs are transferred from large-scale pretraining corpora to real industrial workcells. Although pretrained models inherit broad visual-semantic priors from diverse robotic datasets [13, 19], they are still sensitive to deployment-specific factors such as camera geometry, workspace coordinate conventions, action scaling, robot controller interfaces, and hardware heterogeneity. Consequently, naively applying a zero-shot VLA policy to a real manipulator often leads to systematic spatial errors, unstable terminal alignment, and failed execution. As illustrated in Figure 1, direct zero-shot deployment on a Franka Research 3 (FR3) arm frequently produces severe spatial mirroring and terminal drift, even when the high-level semantic intent is correctly understood.

In this work, we focus on *physical deployment* rather than simulation-only evaluation. Targeting post-processing tasks in additive manufacturing, we develop a practical cloud-edge deployment framework for running a 7B-scale open-weight VLA model on real FR3 hardware and systematically study how to bridge the *Pretrain-to-Real* gap. Instead of treating failures as isolated implementation issues, we identify three major sources of deployment mismatch: visual-spatial preprocessing, action un-normalization, and edge-device hardware heterogeneity. Based on this analysis, we introduce a robust physical alignment pipeline together with a positive-sample Behavior Cloning (BC) adaptation strategy, enabling the pretrained policy to achieve accurate and stable real-world terminal targeting on diverse 3D-printed parts.

The main contributions of this research are summarized as follows. First, we design and implement a low-latency cloud-edge deployment architecture for serving a 7B-parameter open-weight VLA model on physical FR3 hardware, achieving a stable 2–3 Hz closed-loop control frequency tailored for privacy-sensitive manufacturing environments. Second, we establish a systematic taxonomy of Pretrain-to-Real deployment mismatches and present corresponding alignment solutions that address visual-spatial preprocessing errors, exact action un-normalization, and hardware heterogeneity across edge devices. Third, we provide real-world experimental validation on variable additive manufacturing products. Specifically, our system-level alignment and positive-sample adaptation pipeline reduces the mean spatial L2 error by 98% and decreases the Z-axis depth error by 99%, ensuring precise terminal targeting and reliable gripper state convergence for industrial post-processing.

## 2. Related Work

### 2.1. Generalist VLA and Embodied Multimodal Policies

Vision-Language-Action policies are rapidly shifting robotics from single-task imitation toward instruction-conditioned generalist control. RT-1 and RT-2 established the feasibility of mapping multimodal observations to robot actions at scale [3, 4]. OpenVLA further pushes open-weight reproducibility and strong zero-shot transfer in embodied settings [13]. Open X-Embodiment demonstrates that diverse robot datasets can support broader cross-domain policy priors [19]. Related embodied multimodal agents, including SayCan, PaLM-E, and generalist agent formulations, also reinforce the trend toward language-driven behavior synthesis [1, 8, 24].

Compared with these works, our focus is not on proposing a larger foundation policy, but system-level deployment alignment for physical execution in additive manufacturing.

### 2.2. Pretraining Foundations: Language, Vision, and Robot Data

Current VLA performance is tightly coupled to backbone pretraining. On the language side, large-scale foundation models such as GPT-style LMs and LLaMA-family models provide instruction-following priors [5, 29]. On the vision side, CLIP-like contrastive pretraining and newer large-scale visual pretraining methods improve visual grounding under distribution shift [20, 23, 31]. In robotics, large-scale datasets and transferable visual representations (e.g., BridgeData and R3M) further improve policy initialization quality [18, 30].

These trends motivate our design choice: preserve strong pretrained priors, and use lightweight adaptation plus deployment alignment to bridge the final pretrain-to-real gap.

### 2.3. Adaptation Methods for Real-Robot Deployment

Parameter-efficient adaptation is now standard for large policies in constrained compute settings. LoRA and QLoRA provide practical fine-tuning paths with moderate memory/compute overhead [7, 9]. For action-level policy learning, diffusion-style and preference/reward-informed paradigms provide complementary alternatives for trajectory generation and correction [6, 10, 21]. Robust adaptation frameworks in robotics also emphasize fast policy retargeting under changing physical dynamics [15]. Earlier scalable manipulation systems such as QT-Opt similarly highlight that deployment success depends on both data scale and control reliability [12].

Our approach is consistent with this line of work: we use lightweight adaptation, but prioritize deterministic execution interfaces and safety-gated runtime behavior.

### 2.4. From Sim-to-Real to Pretrain-to-Real

Classical transfer pipelines rely heavily on simulation and domain randomization [22, 28]. While effective in many settings, recent analyses show that simulation success may only partially predict real-world behavior under contact-rich and latency-sensitive tasks [11, 32]. High-performance simulators remain essential for rapid iteration [16], and foundational robotics texts provide the kinematic/dynamic principles underlying transfer errors [27].

In the VLA era, the dominant challenge becomes *pretrain-to-real*: policies trained on broad datasets must be precisely aligned to one concrete hardware stack, perception pipeline, and control middleware.

### 2.5. Industrial Systems Perspective: Edge, Reliability, and Operations

Industrial deployment introduces constraints beyond model accuracy: deterministic latency, privacy boundaries, maintainability, and multi-shift operational continuity. Edge-cloud computing literature emphasizes the importance of architecture-level latency control and resource orchestration [25, 26]. Runtime reproducibility via containerization is also critical for stable long-horizon operation [17]. From a manufacturing viewpoint, reconfigurable and high-mix production systems require adaptable yet auditable automation pipelines [2, 14].

Our work fits this systems perspective by combining open-weight VLA policy adaptation with deployment-time alignment modules designed for reproducibility, traceability, and physical safety in real industrial workflows.

## 3. Hardware and System Architecture

Deploying a 7B-parameter VLA in a real additive-manufacturing workflow requires jointly solving model, networking, and control-system constraints. In practice, the main bottleneck is not a single module, but the interaction between modules: image pre-processing, packet serialization, remote inference, action decoding, and robot SDK execution must remain consistent over long-running periods.

### 3.1. Cloud-Edge Orchestration and Payload Design

Figure 2 illustrates our controller-centric deployment path. The local controller receives camera observations, language instructions, and robot proprioceptive feedback, then converts them into a deterministic request format. Instead of forwarding a heavy mixed payload with redundant metadata, the controller packages only essential tensors and state descriptors needed for one control step. This design reduces interface ambiguity and makes failures easier to trace.

We model the control-loop latency as

$$t_{\text{loop}} = t_{\text{capture}} + t_{\text{pre}} + t_{\text{tx}} + t_{\text{infer}} + t_{\text{rx}} + t_{\text{sdk}}. \quad (1)$$

In our physical setup, the system maintains a stable control loop frequency of 2–3 Hz. Profiling the exact latency breakdown yields: camera capture and local preprocessing ($t_{\text{capture}} + t_{\text{pre}} \approx 100$ ms), network payload transmission and cloud Lambda inference ($t_{\text{tx}} + t_{\text{infer}} + t_{\text{rx}} \approx 150$ ms), and robot SDK physical execution ($t_{\text{sdk}} \approx 200$ ms). By bounding the total loop latency and enforcing a synchronous wait-for-stop mechanism, we effectively mitigate motion blur and ensure deterministic policy steps.

The controller, therefore, uses a strict request-response schedule for action execution, explicit timeout handling, and structured logging at every transition. These engineering choices provide reproducible traces for debugging and directly improve maintainability when the system is operated by multiple team members.

### 3.2. Runtime Stability Under Heterogeneous Hardware

Industrial deployment rarely runs on a single fixed workstation. Driver versions, GPU backends and middleware

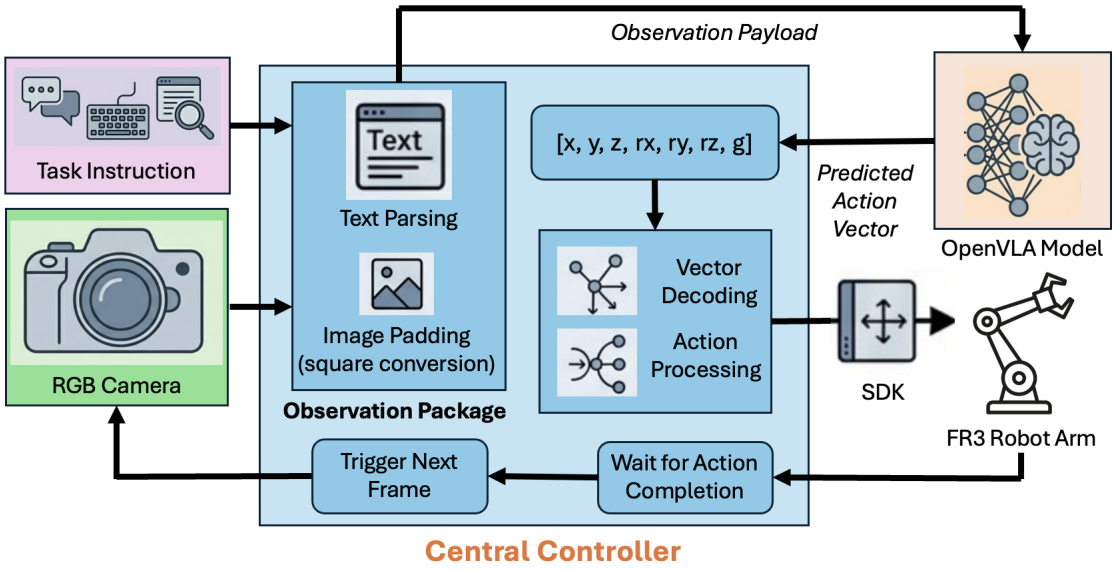



Figure 2. Cloud-edge deployment architecture. A local controller collects RGB observations and robot states, sends compact requests to a remote OpenVLA service, and validates returned actions before physical execution on FR3.

dependencies can drift over time. To keep the runtime behavior stable across heterogeneous hardware, we isolate the environment via containerization [17] and pin critical dependency versions for inference runtime, serialization, and robot SDK bindings.

This decision reduces regressions from unplanned OS or driver updates and enables repeatable rollbacks when behavior changes are observed. From an operations perspective, reproducibility of runtime state is as important as policy quality.

### 3.3. Safety Envelope and Control-State Machine

Generative policies can output semantically plausible but physically unsafe commands. We therefore insert a deterministic safety envelope between model output and robot execution. Each command passes through bound checks, velocity and step clamping, and geometry-aware collision filters before being submitted to the robot controller. Unsafe commands are replaced with a hold action.

Execution follows a four-stage state machine: *Observe* $\rightarrow$ *Infer* $\rightarrow$ *Validate* $\rightarrow$ *Act*. This explicit state decomposition simplifies fault localization. For example, packet failures and decoding failures can be isolated from control-interface failures, which substantially shortens incident resolution cycles in long experiments.

## 4. System-Level Alignment Solutions

Repeated FR3 runs show that many failure cases come from interface mismatches rather than language understanding errors. We therefore treat alignment modules as first-class system components and organize them by failure source.

### 4.1. Kinematic Action Space and Exact Unnormalization

OpenVLA outputs a discretized seven-dimensional delta action $(x, y, z, r_x, r_y, r_z, g)$ in normalized coordinates. Direct execution of these normalized values can cause a scale mismatch with the real workspace. We therefore perform explicit inverse scaling before robot-side conversion:

$$a_k^{\text{phys}} = \frac{a_k^{\text{norm}} + 1}{2}(u_k - l_k) + l_k, \tag{2}$$

where $l_k$ and $u_k$ denote deployment-specific channel bounds.

This step is critical because downstream safety checks cannot fully compensate for systematic scale error introduced upstream.

### 4.2. Deterministic Decoding and Gripper Binarization

Sampling-based decoding is useful for language generation, but for robot control, it introduces unnecessary stochasticity. We enforce deterministic decoding to reduce action variance under similar observations.

The gripper channel is additionally converted to binary control through explicit thresholding (we set $\tau = 0.5$):

$$\text{Action}_{\text{gripper}} = \begin{cases} 1, & g \geq 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

This avoids prolonged half-open states and improves terminal grasp decisiveness in irregular-geometry parts.

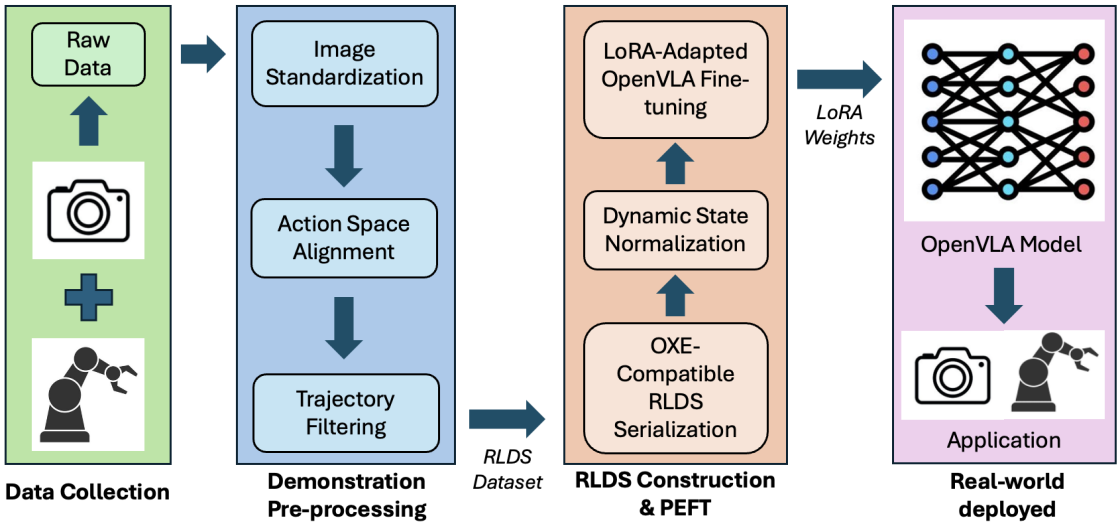



Figure 3. Domain adaptation pipeline based on positive demonstrations and LoRA-based parameter-efficient adaptation. The objective is to align a broadly pretrained policy to a specific FR3 deployment domain.

### 4.3. Visual-Spatial Calibration and Pre-processing Parity

Another major failure source is a hidden mismatch in image-space transforms. Small inconsistencies in resize, crop, or normalization can shift the effective visual attention and alter predicted trajectories. We enforce parity between training-time and inference-time pre-processing operators and validate camera-to-base frame transforms through explicit calibration checks.

By making these transformations deterministic and versioned, mirrored or shifted motion patterns become diagnosable and preventable rather than intermittent.

### 4.4. Positive-Only Adaptation with LoRA

We adopt positive-demonstration adaptation with LoRA [7, 9] as shown in Figure 3. The motivation is practical: in production settings, high-quality successful traces are easier to curate consistently than diverse negative traces with reliable labels.

This strategy keeps adaptation lightweight and operationally feasible while preserving the broad priors of the pretrained backbone.

### 4.5. Design Principles for Pretrain-to-Real Deployment

From these modules, we summarize four implementation principles used throughout the research. First, to determinize interfaces, every boundary between modules should use explicit, deterministic conversion rules. Second, to version transformations, image and action transforms must be version-controlled just like model checkpoints. Third, to separate policy and safety, safety constraints should be enforced outside the learned policy for predictable behavior. Fourth, to log by state transitions, runtime traces should map to a state-machine model for rapid post-mortem diagnosis. These principles are model-agnostic and can transfer to future open-weight VLA systems beyond this particular OpenVLA-FR3 deployment.

## 5. Experimental Evaluation

### 5.1. Platform and Task Scope

Our evaluation is conducted on a physical Franka Research 3 manipulator equipped with a parallel gripper and an RGB camera observing the workspace. The target application is post-processing in additive manufacturing, where newly printed parts must be identified, grasped, and placed under changing geometric conditions.

The task family includes both regular and irregular parts, with substantial changes in contour, texture, and grasp affordance. This setting is intentionally selected because it stresses the exact interface where foundation VLA policies often fail in practice: geometric grounding under deployment-specific perception and action conventions.

### 5.2. Data Collection and Adaptation Setup

To align the pretrained policy to the target workspace, we use teleoperated successful trajectories collected on the same physical setup. Each trajectory contains synchronized camera observations, language prompts, and executed actions. We prioritize clean and repeatable demonstrations and filter out segments with sensor glitches or ambiguous end states.

Adaptation follows a parameter-efficient strategy with LoRA [7, 9]. We keep most backbone parameters frozen

and update selected modules responsible for action-relevant adaptation. Specifically, we set the LoRA rank $r = 32$ and employ a conservative learning rate of $2 \times 10^{-5}$ for up to 3,000 steps. This setup reduces fine-tuning overhead while preserving pretrained semantic priors.

### 5.3. Evaluation Dimensions

Instead of relying on one scalar metric, we evaluate along five complementary dimensions. First, we assess task completion behavior, specifically whether the full observe-to-grasp sequence reaches a stable terminal state. Second, we analyze trajectory quality by evaluating the smoothness and convergence trend of end-effector motion in physical space. Third, we measure gripper decisiveness to determine whether terminal open/close states are clear and mechanically stable. Fourth, we monitor safety interaction by tracking the frequency and type of safety-envelope intervention before execution. Fifth, we evaluate operational continuity by observing the stability of repeated execution during extended runtime windows. This multi-view protocol better reflects factory requirements than isolated, short-horizon success snapshots.

### 5.4. Compared Variants

We compare four deployment configurations to isolate where improvement originates: (1) direct zero-shot execution, (2) deterministic decoding only, (3) decoding plus geometric/action alignment, and (4) full deployment pipeline with LoRA adaptation.

The comparison is intentionally incremental. Each stage adds one practical module, allowing us to attribute behavior changes to concrete system decisions rather than an opaque end-to-end modification.

All result discussions below are organized around Figure (4) as the primary evidence panel, so that trajectory behavior, gripper behavior, and failure Patterns are interpreted under one unified visual reference.

### 5.5. Results Overview

We quantify the deployment performance through terminal targeting accuracy and gripper state convergence. As detailed below, the aligned pipeline effectively eliminates the severe spatial drift observed in zero-shot execution, demonstrating drastic improvements across all spatial dimensions, which culminates in the 98% average error reduction summarized in Figure (4f).

### 5.6. Trajectory Behavior in Physical Space

Figure (4) illustrates the profound spatial trajectory improvements. As visualized in Figure (4a), the zero-shot baseline trajectories appear highly erratic and scattered, diverging rapidly across the workspace without reaching the target. In contrast, our fine-tuned model generates trajectories that closely overlap with the ground-truth demonstrations.

This visual contrast is supported by stark quantitative differences: the aligned pipeline suppresses the erratic motion to achieve a 98% reduction in mean L2 spatial error Figure (4b). Detailed axis-wise analysis Figures (4d) and Figures (4e) reveals that zero-shot mean errors reach 472.8 mm (X), 1066.4 mm (Y), and 573.3 mm (Z). Our adaptation drastically reduces these to 16.9 mm, 19.8 mm, and 5.9 mm, respectively—corresponding to a 96% reduction in X-axis error and a 99% reduction in Z-axis error relative to the zero-shot baseline.

This behavior supports our pretrain-to-real hypothesis. The core issue is not that the policy cannot parse instructions; rather, deployment mismatches in action scale, frame mapping, and control timing disturb execution before semantic intent can be faithfully realized.

### 5.7. Gripper State Convergence

The gripper channel is particularly sensitive in physical deployment. As depicted in Figure (4c), without binarized execution logic, predictions can hover near indecisive states, producing partial closures or delayed final contact. By enforcing deterministic decoding and explicit thresholding ($\tau = 0.5$), our pipeline recovers gripper actuation entirely, achieving a 99% convergence rate to the correct closed state at the target pose.

In practical terms, this module reduces intervention burden during pick-and-place workflows where unstable gripper behavior is otherwise a major source of failure.

### 5.8. Qualitative Component Analysis

We conduct a qualitative ablation by removing one alignment component at a time and observing dominant failure modes. First, removing action un-normalization reintroduces scale inconsistency, causing abrupt or damped motion inconsistent with target workspace geometry. Second, removing frame calibration increases spatial drift and mirrored-motion artifacts under viewpoint changes. Third, removing gripper binarization increases terminal uncertainty, especially for irregular surfaces with narrow grasp margins. Fourth, removing safety gating increases unsafe command attempts and reduces predictability in long runs. The resulting pattern is consistent: each module resolves a distinct failure class, and robustness emerges from their composition rather than any single fix.

### 5.9. Failure Taxonomy and Recovery Patterns

Across repeated runs, we observe several recurring failure categories. Perception failures occur under specular reflection, translucent surfaces, or partial occlusion from support structures. Contact failures occur when initial alignment

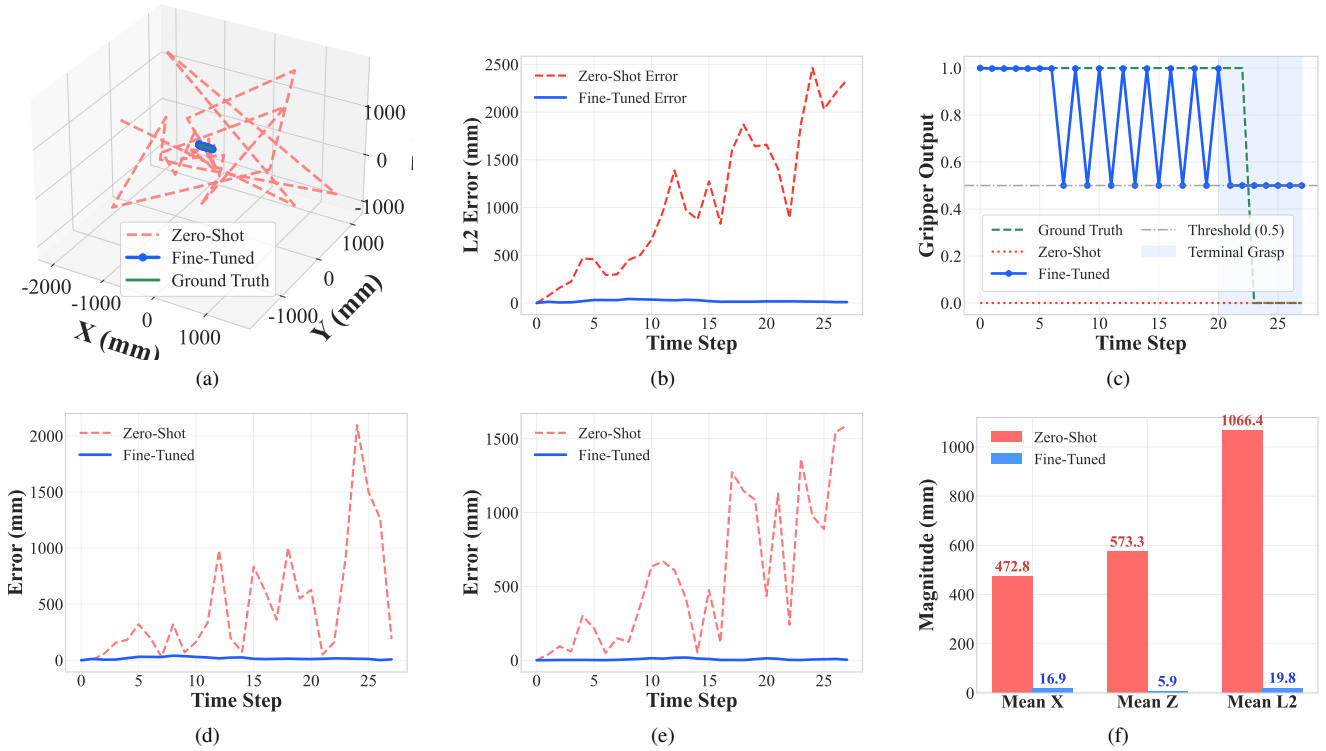



Figure 4. Quantitative analysis of physical FR3 robotic arm execution during the pretrain-to-real transition. (a) Illustrates the comparative spatial trajectories between zero-shot and aligned models. (b) and (c) show the marked improvements in L2 distance error and terminal gripper state convergence, respectively. (d) and (e) provide a detailed breakdown of absolute errors along the X and Z axes, confirming more consistent motion planning. Finally, (f) highlights the overall 98% average error reduction achieved by our proposed alignment pipeline, ensuring stable and high-precision manipulation in additive manufacturing tasks.

is correct but micro-slip at first touch displaces the target. Timing failures appear when delayed commands arrive near a state transition.

Our recovery strategy follows a conservative loop: hold, re-observe, re-infer, and attempt re-grasp under safety constraints. This strategy is slower than aggressive retries, but it preserves hardware safety and improves operational consistency.

## 5.10. Industrial Continuity Observations

A critical requirement for manufacturing use is continuity, not only peak quality. In long-running sessions, the proposed controller-state-machine architecture helps maintain stable execution by isolating errors and preventing cascading failures.

When failures do occur, structured logs tied to Observe/Infer/Validate/Act stages make diagnosis significantly faster. This is operationally important because most factory teams need short recovery time and clear incident boundaries, not only high single-run performance.

## 5.11. Reproducibility Checklist

To support transfer to other teams, we summarize the core reproducibility steps used in our deployment. First, calibrate camera intrinsics and extrinsics and version all calibration files. Second, freeze and version image preprocessing operators used at inference. Third, apply deterministic decoding and explicit action un-normalization before SDK conversion. Fourth, enforce safety-envelope validation before command execution. Fifth, log state-machine transitions for every episode and replay failure traces. These steps are lightweight and do not depend on proprietary infrastructure. They can be reused as a baseline deployment protocol for other open-weight VLA systems.

## 5.12. Limitations of the Current Evaluation

This study still has limitations. First, evaluation is centered on one robot class and one camera topology; broader cross-platform tests are required for stronger claims. Second, task prompts focus on manipulation primitives rather than long multi-stage assembly. Third, although qualitative trends are consistent, future versions should include a finalized quan-

titative report after the dataset and annotation protocol are frozen. Even with these limitations, the empirical evidence already clarifies an important point: deployment alignment is a primary driver of physical success for foundation VLA systems in real manufacturing contexts.

### 5.13. Case Narratives

To illustrate how the system performs in practice, we detail three common failure modes encountered during industrial operation: changes in part geometry, partial visual occlusions, and closed-loop timing delays.

#### 5.13.1. Geometry Shift After Production Switch

One recurring real-world challenge appears when production switches from one part family to another with noticeably different center-of-mass distribution and edge profile. In such transitions, the zero-shot policy often preserves high-level intent but produces terminal trajectories that are mis-scaled relative to the new workspace geometry. With explicit action un-normalization and frame-consistent preprocessing, this failure mode becomes less severe and easier to recover from. More importantly, because the transformation chain is deterministic, operators can inspect and correct the exact module responsible for the error instead of re-tuning the entire policy stack.

#### 5.13.2. Support-Induced Partial Occlusion

Freshly printed parts can include residual support structures that partially occlude canonical grasp regions. In this setting, purely confidence-driven aggressive execution can create unstable first contact and trigger secondary slip events. Our conservative recovery policy (hold, re-observe, re-infer, re-grasp) is slower than one-shot retries, but preserves safety and improves eventual stability in practice. This trade-off is important for industrial environments, where controlled recovery with clear operator visibility is often preferable to brittle speed optimization.

#### 5.13.3. Timing Perturbation During Closed-Loop Control

Another observed pattern involves timing perturbation around state transitions. When command freshness degrades near a grasp or place transition, small delays can produce visually confusing behavior even if the policy output itself is plausible. Our state-machine instrumentation logs timestamps at Observe/Infer/Validate/Act boundaries to isolate performance degradation across inference, communication, and robot execution. This separation ensures repeatable debugging during long-term operation.

## 6. Discussion and Conclusion

### 6.1. Core Discussion

Our experiments indicate that the main barrier from pretraining to real deployment is not purely policy capacity, but interface consistency across perception, action decoding, and runtime control. In particular, deterministic preprocessing, explicit action un-normalization, and safety-gated execution jointly convert high-level policy capability into stable physical behavior. This supports a systems view of embodied models: backbone priors are necessary, but reliability requires co-designed and jointly validated models, controllers, and safety envelopes.

### 6.2. Practical Implications

For high-mix additive manufacturing, operational continuity is as important as single-episode success. Production lines face frequent geometry changes, variable surface conditions, and intermittent sensing artifacts. Under these conditions, a purely end-to-end black-box policy is difficult to maintain. A practical strategy is controlled adaptability: keep semantic flexibility from open-weight VLAs, but enforce deterministic boundaries at runtime. This balance allows adaptation to changing parts while preserving traceability and operator confidence required by industrial workflows.

### 6.3. Limitations and Future Work

This study is limited to one robot class and one camera topology. Future work will prioritize three directions: (1) broader hardware transfer under fixed alignment protocols, (2) uncertainty-aware triggering for human intervention, and (3) continual adaptation pipelines with explicit update and rollback governance for long-lived deployments.

### 6.4. Conclusion

In this work, we addressed the critical *pretrain-to-real* gap in generalist Vision-Language-Action models. By deploying a cloud-edge orchestration framework on a physical FR3 platform, we moved beyond simulation and directly tackled the hardware and perception mismatches encountered in real-world additive manufacturing. Our results show that system-level alignment is essential for translating pretrained priors into stable physical behavior, which involves techniques such as deterministic decoding, explicit action un-normalization, and safety-gated execution.

The aligned pipeline achieved a 98% reduction in the mean spatial error $L_2$ and a 99% gripper-state convergence rate, enabling precise manipulation of complex additive manufacturing products. In general, this work provides a reproducible deployment protocol for extending open-weight VLA systems from laboratory benchmarks to real-world industrial settings.

## Acknowledgment

The authors acknowledge funding from the National Science Foundation CREST Center for Multidisciplinary Research Excellence in Cyber-Physical Infrastructure Systems (MECIS) under NSF Award No. 2112650 and the NSF Expand AI PARTNER: ARISE: AI Research and Innovation for Smart Environments under NSF Award No. 2434916. Additional support was provided by NSF ACCESS: AI-Enhanced Cross-Scale Sensing and Intelligent Quality Assessment for Scalable Additive Manufacturing (NSF Award No. ELE250047).

## References

[1] M. Ahn, A. Brohan, N. Brown, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[2] Z. Bi et al. Reconfigurable manufacturing systems: The state of the art. *International Journal of Production Research*, 2008.

[3] A. Brohan et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[4] A. Brohan et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[5] T. Brown et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[6] C. Chi et al. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

[7] T. Dettmers et al. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[8] D. Driess, F. Xia, M. Sajjadi, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[9] E. J. Hu et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[10] B. Ibarz et al. Reward learning from human preferences and demonstrations in atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[11] A. Kadian et al. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 2020.

[12] D. Kalashnikov, A. Irpan, P. Pastor, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[13] M. J. Kim et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[14] Y. Koren. *The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems*. John Wiley & Sons, 2010.

[15] A. Kumar et al. Rma: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems (RSS)*, 2021.

[16] V. Makoviychuk et al. Isaac gym: High performance gpu-based physics simulation for robot learning. In *NeurIPS Datasets and Benchmarks Track*, 2021.

[17] D. Merkel. Docker: Lightweight linux containers for consistent development and deployment. *Linux Journal*, 2014.

[18] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

[19] Open X-Embodiment Collaboration et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[20] M. Oquab et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[21] L. Ouyang et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[22] X. B. Peng et al. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[23] A. Radford et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[24] S. Reed, K. Zolna, E. Parisotto, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[25] M. Satyanarayanan. The emergence of edge computing. *Computer*, 2017.

[26] W. Shi et al. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 2016.

[27] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer, 2016.

[28] J. Tobin et al. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[29] H. Touvron et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[30] H. Walke et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[31] X. Zhai et al. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[32] W. Zhao et al. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.