FROM ROLLOUTS TO FAILURE MODES: BEHAVIORAL DIAGNOSIS OF VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Task success is the standard measure for evaluating Vision-Language-Action (VLA) models, but it provides limited insight into the failures underlying aggregate performance. We introduce **embodied failure attribution**, a framework that extends success-rate evaluation with behavioral diagnosis. Using Bayesian inference over stage-specific physical interaction evidence from closed-loop rollouts, the framework assigns posterior probabilities to Planning, Grounding, and Action (PGA) failure modes without requiring access to model internals. This provides a common diagnostic representation for **characterizing failure structure, tracking its evolution, and informing targeted interventions**. Across diverse VLA architectures and task settings, we find that Action is a recurring source of failure, while the relative contributions of PGA failure modes vary across models and tasks. During training, the framework distinguishes improvements in overall performance from persistent failure modes: Qwen3VL-PI achieves higher task completion on CALVIN, yet **Action remains the dominant failure dimension at every evaluated checkpoint**, while Grounding attribution declines overall and Planning attribution fluctuates. Under unseen-object conditions, attribution identifies Action as the dominant failure mode, providing additional insight into the observed generalization failures. Finally, an intervention study illustrates how diagnosis can inform model improvement. Action-level correction raises success on an Action-limited task from 51% to 61%, but lowers success on a task with a different failure profile from 65% to 55%, suggesting that the benefit of a correction depends on the failure it addresses. Together, these results establish **behavioral failure attribution as a diagnostic complement to aggregate performance evaluation**, supporting the analysis of VLA systems across model comparison, training, and generalization, while providing a basis for targeted improvement.

1 INTRODUCTION

Vision-Language-Action (VLA) models are typically evaluated by task success, but this coarse measure provides limited insight into the failures underlying overall performance. A failed rollout may reflect an error in task planning, a mismatch between the intended target and its physical grounding, or unsuccessful action execution. Task success collapses these distinct failure modes into the same outcome, $\texttt{success} = 0$, despite their different implications for diagnosis and improvement. Likewise, an increase in success rate does not reveal which failure modes have diminished and which remain unresolved. This motivates a central question: **can physical interaction evidence from closed-loop rollouts support a common functional diagnosis of VLA failures?**

Recent work has moved beyond binary task success through process-aware evaluation and capability-oriented diagnostics, providing finer-grained evidence about behavioral progress and competencies under selected conditions or interventions Ji et al. (2026); Xia et al. (2026); Zhang et al. (2026a); Xu et al. (2026a); Gao et al. (2026). Trajectory-attribution methods more directly associate manipulation failures with predefined failure modes or system components Xu et al. (2026c). Mechanistic and action-level analyses offer complementary insights into internal pathways and behavioral signatures Zhang et al. (2026b); Grant et al. (2026); Gupta (2026). We pursue a complementary goal: inferring functionally defined failure dimensions from observed physical interactions,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

while retaining uncertainty and supporting comparisons across policies with different internal architectures.

We introduce **embodied failure attribution**, a framework that diagnoses Planning, Grounding, and Action (PGA) failures without requiring access to model internals. Planning specifies the intended target, task relation, and operation. Grounding associates that intention with the correct physical instance and interaction region. Action realizes the intended contact and state change through motion and control. Given an instruction and a closed-loop rollout, we organize execution into ordered, target-centered interaction stages and extract evidence about target selection, robot–object geometry, contact, motion, object state, and task-state change. Because evidence becomes diagnostically applicable at different stages, we use event gates to determine which observations enter Bayesian inference. Each PGA dimension is represented by a binary latent variable, yielding a posterior over eight joint states and a marginal failure probability $p(Z_c = 0 \mid E)$ for each dimension c. This formulation retains uncertainty, permits multiple contributing failures, and provides a shared representation of behavioral failure across models and evaluation settings.

We study six policies spanning two training regimes. The first uses a common Qwen3-VL-4B backbone without VLA pretraining: Qwen3VL-4B-GR00T, Qwen3VL-4B-PI, and Qwen3VL-4B-OFT Bai et al. (2025); Ye et al. (2026); Kim et al. (2025). The second comprises pretrained VLAs: GR00T-N1.7, $\pi_{0.5}$, and OpenVLA NVIDIA et al. (2025); Intelligence et al. (2025); Kim et al. (2024). Our experiments span SimplerEnv, CALVIN ABC-D, and RoboTwin, together with the structured L0–L2 UnseenObjects shift from VLA-Arena Zhang et al. (2026a). We organize our findings around three uses of diagnosis: **characterizing failure structure, tracking its evolution, and informing targeted interventions**. First, Action is a recurring failure dimension, but PGA profiles vary across tasks and policies, including those sharing the same vision-language backbone. Second, improved task performance can coexist with persistent failure modes: during Qwen3VL-PI training on CALVIN, task completion improves, yet Action remains the dominant failure dimension at every evaluated checkpoint. Grounding attribution declines overall, while Planning attribution fluctuates. Under unseen-object conditions, Action also dominates the evaluated failure profiles. These results characterize remaining failures rather than directly measuring absolute capability. Finally, action-level correction improves success on the Action-limited Carrot task from 51% to 61%, but reduces success on Stack Cube, which exhibits a different failure profile, from 65% to 55%. This contrast provides initial evidence that diagnosis can inform the selection of interventions.

Our contributions are:

- **A behavioral failure attribution framework.** We use gated Bayesian inference to convert stage-specific physical interaction evidence into a joint PGA posterior, retaining uncertainty and multiple contributing failures without accessing model internals.
- **A shared diagnostic representation for VLA evaluation.** We characterize failure structure across architectures and tasks and track its evolution during training and under distribution shift, exposing distinctions hidden by aggregate success.
- **An empirical connection between diagnosis and intervention.** We study action-level correction across tasks with different failure profiles, providing initial evidence for diagnosis-informed model improvement.

2 Related Work

Existing work related to VLA diagnosis studies different levels of the same problem: how to move beyond terminal task success and understand why an embodied policy succeeds or fails. We organize prior work into three lines: diagnostic evaluation of capabilities, failure attribution in embodied manipulation, and mechanistic or behavioral analysis of VLA policies.

Diagnostic Evaluation of VLA Capabilities. MetaFine decomposes manipulation competency into understanding, perception, and controlled behavior, and evaluates these competencies through decomposed scenarios and semantic, geometric, and photometric interventions Xu et al. (2026b). EBench and VLA-Arena organize tasks and distribution conditions to expose capability and robustness differences across policies Gao et al. (2026); Zhang et al. (2026a). Process-aware evaluators further augment terminal outcomes with dense judgments of progress, process quality, and outcome

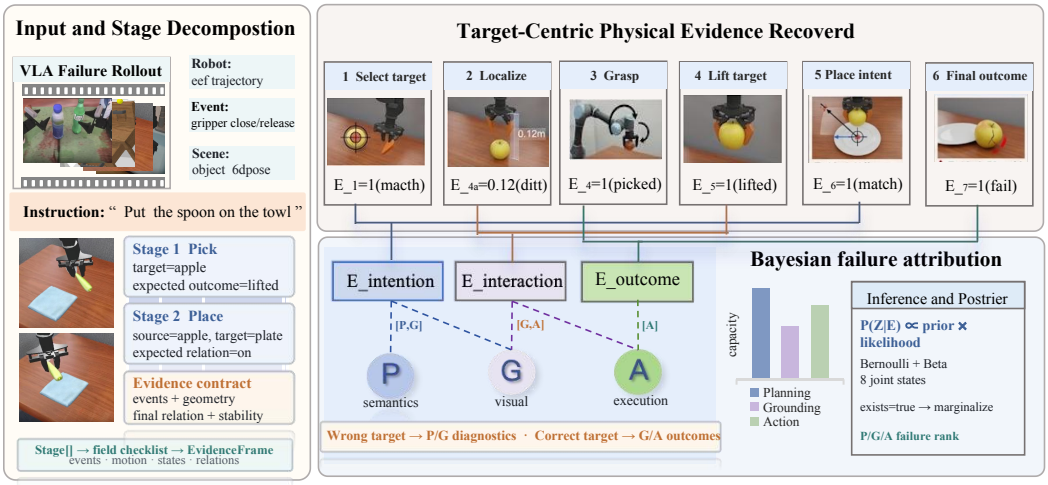


Figure 1: Overview of the physical-interaction attribution framework.

Ji et al. (2026), while RoboProcessBench evaluates VLM-side understanding of phase, contact, motion, temporal order, and primitive transitions Xia et al. (2026). These approaches provide richer measurements of behavioral differences and capability deficiencies, but they typically assess predefined capabilities through designed tasks, interventions, or observer-side measurements.

Failure Attribution in Embodied Manipulation. GTP-FA analyzes failed manipulation trajectories and distinguishes grasp-related failures, such as grasp-functional mismatch and grasp instability, from downstream planning and control failures, enabling diagnosis-guided optimization Xu et al. (2026a). However, their attribution space is defined by the underlying manipulation pipeline: the diagnosis asks which predefined component or failure mode is responsible within a particular system decomposition. This differs from inferring a shared functional capability space directly from the physical evidence produced during execution. Our framework instead defines Planning, Grounding, and Action as external functional capabilities and attributes failures to these capabilities without assuming a particular internal module decomposition.

Mechanistic and Behavioral Analysis of VLA. Another line of work seeks to explain embodied behavior through model internals or observable action signatures. Mechanistic studies probe representations, visual and language features, and causal pathways involved in action generation Li et al. (2026); Lu et al. (2025); Häon et al. (2025). Embodied Interpretability uses interventional visual–action attribution to study causal influence and nuisance features, while Not All Features Are Created Equal and VLA-Trace analyze representations, pathways, and behavior traces across VLA architectures Zhang et al. (2026b); Grant et al. (2026); Shi et al. (2026). Complementary black-box analyses characterize failure through action-level motor signatures Gupta (2026). These approaches reveal internal mechanisms or behavioral patterns associated with VLA performance and failure, but their analysis is tied to model-specific representations, pathways, or action statistics rather than a common functional diagnosis of what limited task completion in an individual rollout. Our approach provides a unified framework for diagnosing VLA capability bottlenecks by inferring failure causes from physical evidence in naturally occurring failed rollouts.

3 METHOD

3.1 PROBLEM FORMULATION: BEHAVIORAL FAILURE ATTRIBUTION

We formulate embodied failure attribution as inverse inference from observed physical interactions to functionally defined failure dimensions. Given a language instruction l and a failed closed-loop rollout τ, the goal is to infer which dimensions are implicated by the execution evidence and with what uncertainty. The rollout records robot actions and states, contacts, object motions, and task-state changes. Diagnosis uses these observable consequences without requiring access to the policy's internal representations or module structure.

We characterize execution through three functional dimensions: **Planning (P)**, which specifies the intended target, task relation, and operation; **Grounding (G)**, which associates that intention with the correct physical instance and interaction region; and **Action (A)**, which realizes the intended contact and state change through motion and control. These dimensions provide a shared vocabulary for behavioral diagnosis across VLA architectures, rather than assuming that each policy contains separate planning, grounding, and action modules.

Let $Z = (Z_P, Z_G, Z_A) \in \{0,1\}^3$ denote the latent functional state of the observed execution, where $Z_c = 0$ represents a failure in dimension c and $Z_c = 1$ its absence. Multiple dimensions may contribute to the same failure, yielding eight joint states. Given evidence $E = \Phi(l, \tau)$, we report

$$q(\tau) = [p(Z_P = 0 \mid E),\, p(Z_G = 0 \mid E),\, p(Z_A = 0 \mid E)]\,. \tag{1}$$

This profile expresses evidence-based attribution for the observed rollout; it does not directly measure the policy's absolute capabilities or identify a unique internal cause. The central challenge is to construct evidence that distinguishes competing PGA explanations while respecting when each observation becomes diagnostically applicable.

3.2 Stage-Specific Physical Interaction Evidence

Target-centered interaction stages. Physical evidence acquires diagnostic meaning relative to the interaction being attempted. We therefore organize execution into an ordered sequence of target-centered stages $\mathcal{S} = (S_1, \ldots, S_K)$. Each stage specifies a required target or interaction region g_k^*, an intended operation and motion constraint u_k^*, and an expected world-state change Δx_k^*. Its associated trajectory segment τ_k contains the observations available for assessing that interaction. For example, picking and placing an object involves a grasp stage targeting the object and a placement stage specifying the destination and desired spatial relation.

Evidence extraction and event gates. Within each stage, we extract evidence about target selection, interaction geometry, contact, motion, and the resulting physical state. For evidence type j, an extractor Φ_j evaluates the observed interaction against the stage specification. An event gate $R_{k,j} \in \{0,1\}$ indicates whether its prerequisite events have occurred and the required observations are available. The evidence supplied to inference is

$$E(l, \tau) = \{(k, j, e_{k,j}) : R_{k,j} = 1, \quad e_{k,j} = \Phi_j(\tau_k; g_k^*, u_k^*, \Delta x_k^*)\}\,. \tag{2}$$

The stage index preserves each observation's interaction context, while the gates determine which likelihood terms enter inference.

Gating distinguishes an observed failure from an unevaluable outcome. For example, retention becomes evaluable only after an object has been grasped and lifted. If these prerequisite events do not occur, retention is omitted rather than recorded as failed. Likewise, engagement with an incorrect object provides evidence about target selection but does not establish whether the policy could grasp or lift the intended object. The upstream mismatch remains diagnostic; unobserved downstream outcomes contribute no additional evidence.

Diagnostic interpretation. Table 1 summarizes the evidence types and their associated PGA dimensions. Target mismatch may reflect either Planning or Grounding; the incorrect object's task relevance and visual similarity provide additional evidence for distinguishing these explanations. Localization primarily constrains Grounding, whereas contact and goal-state attainment may involve both Grounding and Action. Lift, retention, and post-interaction stability provide evidence about Action once their prerequisites are satisfied.

These associations support probabilistic inference rather than deterministic labeling. Correct target engagement followed by a failed state change, for example, strengthens an Action explanation when localization and contact evidence also support successful grounding. The same interface applies to grasping, placing, pushing, pulling, pressing, and rotating by instantiating their targets, operations, expected changes, and event gates.

3.3 Gated Bayesian Inference

Evidence likelihoods. Individual observations can admit multiple PGA explanations. We therefore maintain a posterior over all eight joint states, rather than assigning each failed rollout a single

Table 1: Physical evidence and its diagnostic parent dimensions. Each item enters inference only when its event gate is active.

Evidence	Diagnostic question	Parents
Target match	Did the policy engage the required physical target?	P, G
Wrong-target context	Is the incorrect target compatible with the task context?	P
Wrong-target similarity	Does the incorrect target visually resemble the required target?	G
Operation/direction	Does the observed operation agree with the required direction or axis?	P, G
Localization	Is the interaction point aligned with the target affordance?	G
Contact/grasp	Was valid physical engagement established?	G, A
Lift/retention	Was the object lifted and retained during transfer?	A
State/relation	Was the required object state, joint state, or task relation achieved?	G, A
Stability	Did the achieved state persist after release or withdrawal?	A

label. Each evidence item has a diagnostic parent set $\mathrm{pa}(k,j) \subseteq \{P, G, A\}$, specifying the dimensions on which its likelihood depends. Binary events use Bernoulli likelihoods, while continuous measurements normalized to $(0, 1)$ use Beta likelihoods:

$$\ell_{k,j}\big(e \mid Z_{\mathrm{pa}(k,j)}\big) = \begin{cases} \mathrm{Bern}\big(e; \theta_{k,j}(Z_{\mathrm{pa}(k,j)})\big), & \text{binary evidence,} \\ \mathrm{Beta}\big(e; \alpha_{k,j}(Z_{\mathrm{pa}(k,j)}), \beta_{k,j}(Z_{\mathrm{pa}(k,j)})\big), & \text{continuous evidence.} \end{cases} \tag{3}$$

The likelihood parameters specify how each observation is distributed under its parent states, and $\pi(z)$ denotes the joint prior over PGA. These parent sets encode diagnostic assumptions about physical interactions, not dependencies between internal policy modules.

Joint posterior. We assume that included evidence items are conditionally independent given the joint PGA state. Combining their likelihoods with the prior gives

$$p(Z = z \mid E) = \frac{\pi(z) \prod_{(k,j):R_{k,j}=1} \ell_{k,j}\big(e_{k,j} \mid z_{\mathrm{pa}(k,j)}\big)}{\sum_{z' \in \{0,1\}^3} \pi(z') \prod_{(k,j):R_{k,j}=1} \ell_{k,j}\Big(e_{k,j} \mid z'_{\mathrm{pa}(k,j)}\Big)}. \tag{4}$$

Only applicable observations contribute likelihood terms. This allows evidence from successive interaction stages to update competing explanations while preserving uncertainty and combinations of contributing failures. Since the state space contains only eight configurations, the posterior can be computed by exact enumeration.

Failure profiles and interpretation. For each dimension $c \in \{P, G, A\}$, we marginalize the joint posterior:

$$q_c(E) = p(Z_c = 0 \mid E) = \sum_{z:z_c=0} p(Z = z \mid E), \qquad q(E) = \big(q_P(E), q_G(E), q_A(E)\big). \tag{5}$$

A larger $q_c(E)$ indicates stronger attribution to dimension c under the evidence model. The marginals describe overlapping failure events, so they need not sum to one.

Averaging these profiles over failed rollouts characterizes the remaining failure structure of a model, task, checkpoint, or evaluation condition. Because this aggregation is conditioned on failure, it should be interpreted alongside task performance: an increase in success rate can coexist with a persistent dominant failure dimension. The resulting profiles support comparisons across policies and settings and identify candidate dimensions for targeted intervention, whose benefits must be established through subsequent evaluation.

4 Experiments

We evaluate the attribution framework through an aggregate model–benchmark comparison, controlled primitive and architecture comparisons, checkpoint and OOD analyses, and a targeted intervention to answer four questions: (1) What capability most often limits a failed rollout? (2) How

Table 2: Native task performance and failure-conditioned PGA profiles. Task performance is success rate (%) for SimplerEnv and RoboTwin, and average completed sequence length for CALVIN ABC-D. P/G/A report mean marginal failure probabilities (%) over failed trajectories; they are not mutually exclusive. A dash denotes a model-condition pair not evaluated under the current protocol.

		Qwen3VL-4B, no VLA pretraining			Pretrained VLA		
Benchmark	Metric	Qwen-GR00T	Qwen-PI	Qwen-OFT	$\pi_{0.5}$	GR00T-N1.7	OpenVLA
SimplerEnv	SR ↑	49.75%	60.25%	45.00%	59.52%	62.50%	N/A‡
	P ↓	19.08	10.26	13.99	18.60	18.32	–
	G ↓	0.08	0.01	0.02	36.61	32.03	–
	A ↓	54.96	47.14	43.97	76.82	70.27	–
CALVIN ABC-D	Avg. length ↑	3.727	3.792	3.873	3.761	3.371	2.681
	P ↓	17.35	10.66	21.99	28.30	30.85	24.20
	G ↓	28.38	29.29	22.96	22.90	17.51	28.50
	A ↓	54.30	71.06	61.00	68.30	61.75	61.70
RoboTwin	SR ↑	24.40%	N/A‡	88.00%	11.00%	9.00%	18.00%
	P ↓	18.49	–	5.37	18.59	18.09	18.49
	G ↓	73.49	–	71.80	95.55	73.97	73.82
	A ↓	76.59	–	96.32	97.39	52.15	63.38

does the failure structure change across task primitives and model implementations? (3) How do capability profiles evolve during training and under distribution shift? (4) Can the inferred profile identify which capability to optimize for a particular task? All reported P/G/A values are failure-conditioned marginal posteriors over the coupled capability state.

4.1 Experimental Setup

Models. The no-VLA-pretraining group contains Qwen3VL-4B-GR00T, Qwen3VL-4B-PI, and Qwen3VL-4B-OFT. They share the same vision-language backbone and training setting, while their action architectures differ. The pretrained-VLA group contains GR00T-N1.7, $\pi_{0.5}$, and OpenVLA. This grouping supplies both a shared-backbone comparison and cross-family model comparisons.

Benchmarks. SimplerEnv covers rigid-body pick-and-place and exposes target selection, grasp, lift, transport, and placement behavior. CALVIN ABC-D adds articulated objects, switches, and multi-stage manipulation and is used for primitive-level analysis. RoboTwin extends the interaction set with the reported bottle-picking tasks. VLA-Arena provides the L0–L2 UnseenObjects shift relative to the L0 post-training data. The benchmark-specific studies use the same physical-interaction observation and PGA attribution interface.

Metrics. Let $S_{m,b}$ denote the native task-performance metric of model m on benchmark b. SimplerEnv and RoboTwin use task success rate, whereas CALVIN uses the ABC-D average completed sequence length. We report each native metric alongside the failure-conditioned PGA profile. For failure attribution, define the set of failed trajectories as

$$\mathcal{F}_{m,b} = \{\tau \mid \tau \text{ is a failed trajectory of } m \text{ on } b\}. \tag{6}$$

For capability $c \in \{P, G, A\}$, the failure-conditioned marginal is

$$\bar{q}^{c}_{m,b} = \frac{1}{|\mathcal{F}_{m,b}|} \sum_{\tau \in \mathcal{F}_{m,b}} p(Z_c = 0 \mid E_\tau). \tag{7}$$

The three quantities $\bar{q}^{P}_{m,b}$, $\bar{q}^{G}_{m,b}$, and $\bar{q}^{A}_{m,b}$ form the model's failure profile under a task condition. They are marginal probabilities rather than proportions of mutually exclusive classes; consequently, their sum is not constrained to one.

4.2 Main capability bottleneck

Table 2 summarizes native task performance and failure-conditioned PGA profiles across the model–benchmark matrix.

Action is the largest marginal in every SimplerEnv row (43.97–76.82%) and every CALVIN row (54.30–71.06%). The same pattern appears in the evaluated RoboTwin profiles: Action reaches 76.59% for Qwen-GR00T, 96.32% for Qwen-OFT, and 97.39% for $\pi_{0.5}$. These values identify

Table 3: GR00T-N1.7 CALVIN ABC-D action-level success and failure-conditioned PGA profiles (%).

Primitive	N/f	SR ↑	P ↓	G ↓	A ↓
Lift	892/117	86.9	61.5	4.5	**96.1**
Rotate	379/24	93.7	35.2	3.7	43.4
Move	470/29	93.8	26.3	24.3	**66.2**
Place	469/115	75.5	17.5	35.7	**70.1**
Push	402/134	66.7	24.5	0.7	**64.9**
Drawer	586/15	97.4	17.3	**74.4**	24.1
Stack	181/137	24.3	24.5	24.2	28.2

physical realization–contact, grasp, lift, retention, transport, and final state change–as the most persistent source of failed rollouts.

The upstream dimensions remain visible in the same table. CALVIN Planning spans 10.66–30.85% and Grounding spans 17.51–29.29% across the reported models, while RoboTwin Grounding reaches 73.49% for Qwen-GR00T, 71.80% for Qwen-OFT, and 95.55% for $\pi_{0.5}$. The aggregate profile is therefore Action-heavy without collapsing the diagnosis into a single failure label.

4.3 Task and architecture effects

The aggregate Action pattern in Table 2 changes when the executed primitive changes. We first fix GR00T-N1.7 and compare CALVIN primitives, then fix the Qwen3VL-4B vision-language backbone and compare action architectures.

Task primitives. Table 3 reports action-level success and failure-conditioned PGA posteriors for GR00T-N1.7 on CALVIN ABC-D. The same policy exhibits distinct profiles across primitives: Lift is 96.1% Action with 86.9% success, Drawer is 74.4% Grounding with 97.4% success, and Rotate is 43.4% Action with 93.7% success. Move and Place remain Action-dominated at 66.2% and 70.1%, while Stack has a flatter profile (24.5% Planning, 24.2% Grounding, and 28.2% Action) together with 24.3% success. The primitive therefore determines which capability is most prominent even within one VLA; high task success does not imply a common failure structure.

Shared-backbone action architectures. With the Qwen3VL-4B backbone and no VLA pretraining held fixed, Qwen-GR00T, Qwen-PI, and Qwen-OFT remain Action-leading but produce different posterior magnitudes. Their Action posteriors are 54.96%, 47.14%, and 43.97% on SimplerEnv, and 54.30%, 71.06%, and 61.00% on CALVIN, respectively. The architecture comparison changes the strength of the Action signal within the same task family, whereas the primitive comparison changes the leading capability itself. Task conditions provide the primary structure of the diagnosis, with action architecture modulating that structure.

4.4 Training and distribution shifts

The capability profile also changes as the policy is trained and evaluated away from its post-training distribution. The following comparisons separate capability acquisition during training from capability preservation under object and task shifts.

Training. Table 4 reports the Qwen3VL-PI training trajectory on CALVIN, with 1,000 evaluated sequences at each checkpoint and 23,977 attempted subtasks overall. For the no-VLA-pretraining Qwen3VL-PI model, the average completed sequence length increases from 2.875 at 10k steps to 3.792 at 60k, while attempted-subtask success rises from 82.05% to 91.07% and 5/5 sequence completion rises from 37.1% to 62.8%. Action remains the largest failure dimension at every checkpoint, decreasing from 63.54% to 57.00%. Grounding declines overall from 23.61% to 14.58%, whereas Planning varies across checkpoints, reaches 24.09% at 50k, and ends at 21.51%. Thus, convergence improves overall task completion while leaving a persistent Action-heavy residual failure profile and changing the relative contribution of the upstream capabilities.

Distribution shift. On the $\pi_{0.5}$ ManiSkill25 split, success falls from 45.0% on post-training tasks to 1.0% on unseen tasks. Grounding rises from 16.64% to 47.71%, while Action remains high at 80.84% and 74.59%, respectively. The shift therefore changes the stage at which the execution chain fails rather than producing a uniform capability collapse.

Table 4: Qwen3VL-PI CALVIN ABC-D training trajectory. Average length is the native CALVIN metric. P/G/A are mean marginal failure probabilities over failed subtasks (%); the endpoint subtask-SR and 5/5 completion values are reported in the text.

Checkpoint	Avg. length ↑	P ↓	G ↓	A ↓
10k	2.875	17.40	23.61	**63.54**
20k	3.643	21.54	25.54	58.89
30k	3.549	17.97	19.43	**58.03**
40k	3.673	20.85	17.34	57.71
50k	3.777	24.09	18.48	58.23
60k	3.792	21.51	14.58	**57.00**

VLA-Arena UnseenObjects provides a larger structured shift from L0 to L1 and L2, defined relative to the L0 post-training data. The level-wise success rates and failure-conditioned PGA profiles are shown in Table 5.

Table 5: VLA-Arena UnseenObjects object-category OOD results over 500 episodes per condition. P/G/A report failure-conditioned mean marginal failure probabilities (%).

Model	Condition	SR ↑	P ↓	G ↓	A ↓
Qwen-GR00T	L0	63.2	18.58	12.08	**79.14**
	L1	48.6	31.08	5.53	84.43
	L2	23.2	8.83	11.14	**90.12**
$\pi_{0.5}$	L0	62.0	18.89	4.32	**78.94**
	L1	47.6	20.31	3.97	80.88
	L2	**19.8**	45.54	35.10	**59.48**

Table 5 shows the same pattern under the VLA-Arena UnseenObjects shift. Relative to L0, success decreases from 63.2% to 48.6% at L1 and 23.2% at L2, while Action increases from 79.14% to 84.43% and 90.12%. L1 also raises Planning from 18.58% to 31.08%, whereas L2 contains a separate target-binding signal: target-match observations are positive in 54.2% of valid L2 failure cases. The rollout can therefore retain upstream target evidence while failing during grounding, lifting, and retention. The success rate alone does not expose this progression.

The same shift produces a different profile for $\pi_{0.5}$. Its success rate decreases from 62.0% at L0 to 47.6% at L1 and 19.8% at L2. L0 and L1 remain Action-dominated, with Action posteriors of 78.94% and 80.88%, respectively. At L2, however, Planning and Grounding rise to 45.54% and 35.10%, while Action decreases to 59.48%. The dominant failed-trajectory joint state is 010, corresponding to normal Planning with Grounding failure and normal Action. The L2 condition has `target_match=0` in 270 of 500 episodes, making the shift from an execution-heavy to a coupled target-binding and execution profile explicit. Comparing the two pretrained policies under the same L0–L2 structure further shows that the nominal object shift does not determine a unique failure mode: the realized profile depends on the policy's interaction behavior.

Together, the training and shift comparisons show that capability profiles change in stages. Training reduces Action and Grounding failures while distribution shifts preserve parts of the upstream process and move failures toward physical binding and execution.

4.5 Task-specific intervention

We use the inferred profile to select an intervention target on SimplerEnv. We freeze the original π_0 policy and apply RL^2-VLA at inference time to generate and filter additional action candidates. The intervention changes action generation and selection while leaving the language and visual inputs unchanged. Table 6 reports transitions from the original outcome to the intervened outcome.

The action intervention improves Carrot from 51% to 61% by repairing 23 of 49 original failures, while damaging 13 of 51 original successes. The same intervention reduces Stack cube from 65% to 55%: it repairs 10 of 35 failures but damages 20 of 65 original successes. Spoon has nearly balanced repair and damage (12 versus 14 episodes), while Eggplant is close to saturation, with 8 of 8 failures repaired and 4 of 92 successes damaged. The task-level contrast shows that the diagnosed profile

Table 6: Action-side intervention on SimplerEnv π_0 episodes. Each entry reports original failures repaired, original successes damaged, and the net success-rate change.

Task	Failure → success	Success → failure	Net change
Eggplant	8/8 (100.0%)	4/92 (4.3%)	+4 pp
Spoon	12/44 (27.3%)	14/56 (25.0%)	−2 pp
Stack cube	10/35 (28.6%)	20/65 (30.8%)	−10 pp
Carrot	23/49 (46.9%)	13/51 (25.5%)	+10 pp

selects where an action-side optimization is useful: it improves the Action-limited Carrot task and harms Stack cube, whose failures involve different stage and placement constraints.

4.6 Recorded-robot physical interaction

The physical-interaction protocol is instantiated on a recorded real-robot rollout using synchronized robot, scene, and task records. Robot-side actions and states are aligned with RGB-D frames, camera intrinsics, and object geometry. FoundationPose tracks object poses and bounding boxes from the scene assets and first-frame masks. Gripper closure and release, contact and grasp outcomes, and the task description are then aligned to form the same stage-scoped EvidenceFrame used by the simulated benchmarks.

For the recorded DROID episode, the task is to pick up a green cube and place it in a bowl. At the first gripper closure, the selected entity matches the task target, but the interaction point is substantially displaced from the target and no valid contact or grasp is observed. The resulting failure marginals are $(P, G, A) = (0.1860, 0.9922, 0.6682)$, with joint state 100 receiving the highest posterior mass. The rollout therefore exhibits a Grounding-dominant failure with a coupled Action component, demonstrating that the intention/outcome evidence interface and PGA posterior can be applied to physical robot execution.

5 Limitations and Future Work

Our framework provides functional diagnosis rather than internal causal explanations, and its reliability depends on event gates and inference assumptions. Future work should evaluate attribution accuracy and posterior calibration with annotated rollouts and controlled fault injection.

Our evaluation is primarily simulated, with a single real-robot case demonstrating the evidence interface. Broader deployment requires robust evidence extraction under imperfect observations. Failure-conditioned profiles also describe residual rather than absolute capability and should be interpreted with task performance.

The intervention study covers one action-level correction on a limited set of tasks. Future work should test interventions targeting different PGA dimensions and evaluate both repaired failures and damaged successes.

6 Conclusion

We introduced embodied failure attribution, a framework that converts stage-specific physical interaction evidence from closed-loop rollouts into a joint posterior over Planning, Grounding, and Action failures. By retaining uncertainty and multiple contributing dimensions without accessing model internals, it provides a shared diagnostic representation across VLA architectures. Our experiments show that this representation can characterize model- and task-dependent failure structure, track residual failures during training and distribution shift, and inform targeted interventions. Action recurs as a dominant failure dimension, while similar changes in task performance can accompany different PGA profiles. The intervention results further suggest that the benefit of action-level correction depends on the failure it addresses. Overall, behavioral failure attribution complements aggregate performance evaluation by providing a capability-level view of why VLA rollouts fail.

AI USE STATEMENT

AI tools assisted with drafting and language editing. All analyses, editorial decisions, and conclusions are the authors' own. The authors reviewed the AI-assisted text and take full responsibility for the manuscript.

REPRODUCIBILITY STATEMENT

The appendix documents the attribution model and gating rules in Probabilistic Attribution Details, the model, benchmark, and evaluation settings in Experimental Setup and Implementation, and the intervention protocol in Attribution-Guided Intervention Details.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Ning Gao, Jinliang Zheng, Xing Gao, Haoxiang Ma, Hanqing Wang, Yukai Wang, Jiantong Chen, Zanxin Chen, Shujie Zhang, Mingda Jia, Xuekun Jiang, Zihou Zhu, Xinyu Li, Shuai Wang, Hao Li, Wenzhe Cai, Yuqiang Yang, Xudong Xu, Zhaoyang Lyu, Yao Mu, Tai Wang, Jiangmiao Pang, Jia Zeng, Weinan Zhang, and Chunhua Shen. Ebench: Elemental diagnosis of generalist mobile manipulation policies, 2026. URL https://arxiv.org/abs/2606.18239.

Bryce Grant, Xijia Zhao, and Peng Wang. Not all features are created equal: A mechanistic study of vision-language-action models, 2026. URL https://arxiv.org/abs/2603.19233.

Krishnam Gupta. How vlas fail differently: Black-box action monitoring reveals architecture-specific failure signatures, 2026. URL https://arxiv.org/abs/2605.28726.

Bear Häon, Kaylene Stocking, Ian Chuang, and Claire Tomlin. Mechanistic interpretability for steering vision-language-action models, 2025. URL https://arxiv.org/abs/2509.00328.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Yuheng Ji, Yuyang Liu, Huajie Tan, Xuchuan Huang, Fanding Huang, Yijie Xu, Cheng Chi, Yuting Zhao, Huaihai Lyu, Peterson Co, Mingyu Cao, Qiongyu Zhang, Zhe Li, Enshen Zhou, Pengwei Wang, Zhongyuan Wang, Shanghang Zhang, and Xiaolong Zheng. Prm-as-a-judge: A dense evaluation paradigm for fine-grained robotic auditing, 2026. URL https://arxiv.org/abs/2603.21669.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL https://arxiv.org/abs/2502.19645.

Chenyang Li, Jieyuan Liu, Bin Li, Bo Gao, Yilin Yuan, Yangfan He, Yuchen Li, and Jingqun Tang. Dtp: A simple yet effective distracting token pruning framework for vision-language action models, 2026. URL https://arxiv.org/abs/2601.16065.

Hong Lu, Hengxu Li, Prithviraj Singh Shahani, Stephanie Herbers, and Matthias Scheutz. Probing a vision-language-action model for symbolic states and integration into a cognitive architecture, 2025. URL https://arxiv.org/abs/2502.04558.

NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

Haoyuan Shi, Xiancong Ren, Yingji Zhang, Qinfan Zhang, Jiayu Hu, Haozhe Shan, Han Dong, Jinpeng Lu, Yinda Chen, Yi Zhang, Yong Dai, and Xiaozhu Ju. Vla-trace: Diagnosing vision-language-action models through representation and behavior tracing, 2026. URL https://arxiv.org/abs/2605.30117.

Dayu Xia, Yue Shi, Yao Mu, Huiting Ji, Chaofan Ma, Yingjie Zhou, Hua Chen, Yang Liu, Jiezhang Cao, and Guangtao Zhai. Roboprocessbench: Benchmarking process-aware understanding in vision-language robotic manipulation, 2026. URL https://arxiv.org/abs/2606.13040.

Haiweng Xu, Sipeng Zheng, Hao Luo, Wanpeng Zhang, Ziheng Xi, and Zongqing Lu. Unmasking the illusion of embodied reasoning in vision-language-action models, 2026a. URL https://arxiv.org/abs/2604.18000.

He-Yang Xu, Pengyuan Zhang, Zongyuan Ge, Xiaoshuai Hao, Serge Belongie, Xin Geng, Yuxin Peng, and Xiu-Shen Wei. Beyond binary success: A diagnostic meta-evaluation framework for fine-grained manipulation, 2026b. URL https://arxiv.org/abs/2605.19986.

Jiahao Xu, Peiyuan Wang, Hanzhuo Zhang, Zihao Yu, Tianyu Fu, Hao Chen, Xuanhao Xiang, Jianbo Yu, Chenchen Fu, and Wanyuan Wang. Grasp-then-plan with failure attribution: A closed two-stage framework for precise and generalizable robotic manipulation, 2026c. URL https://arxiv.org/abs/2606.03385.

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. Starvla-α: Reducing complexity in vision-language-action systems, 2026. URL https://arxiv.org/abs/2604.11757.

Borong Zhang, Jiahao Li, Jiachen Shen, Yuhao Zhang, Yishuai Cai, Lu Liu, Hailu Ji, Yuanpei Chen, Juntao Dai, Jiaming Ji, and Yaodong Yang. Vla-arena: An open-source framework for benchmarking vision-language-action models, 2026a. URL https://arxiv.org/abs/2512.22539.

Hanxin Zhang, Mingshuo Xu, Abdulqader Dhafer, Shigang Yue, Hongbiao Dong, and Zhou Daniel Hao. Embodied interpretability: Linking causal understanding to generalization in vision-language-action models, 2026b. URL https://arxiv.org/abs/2605.00321.

A FUNCTIONAL FAILURE DIMENSIONS AND PHYSICAL EVIDENCE

This appendix makes the evidence interface used by the attribution model explicit. The central unit is a target-centered interaction stage. A stage specifies the physical entity or region that should be interacted with, the operation to be performed, and the state change that should result. The observation layer then records what the policy attempted and what the environment actually underwent. This preserves the chain

$$\text{failed rollout} \longrightarrow \text{physical interaction evidence} \longrightarrow \text{behavioral PGA attribution.}$$

A.1 PGA FUNCTIONAL DEFINITIONS

We use three binary latent functional-state variables,

$$Z = (Z_P, Z_G, Z_A) \in \{0, 1\}^3,$$

where $Z_c = 0$ denotes a failure in dimension c and $Z_c = 1$ its absence in the modeled execution. These variables describe external functional dimensions, not absolute policy competence or internal neural modules. Multiple dimensions may contribute to the same failed rollout, and their posterior probabilities depend on the available physical evidence and the inference assumptions.

Planning. Planning (P) is the capability to interpret the instruction and commit to the required interaction target, task relation, operation, stage, or motion direction. In a pick-and-place task, Planning determines which object should be picked and which relation should hold after placement. In a push, pull, press, slide, or rotate stage, it determines the intended object or part and the operation that should change the world state. Planning answers the functional question "what should be done, and with which task-level relation?"

Grounding. Grounding (G) is the capability to bind the intended semantic target to the correct physical instance, interaction region, and local geometric constraint. It covers selection among visually similar entities, localization of the contact or release point, and the spatial binding needed to realize the instruction. Grounding answers "where should the interaction occur in the scene?"

Action. Action (A) is the capability to realize an intended interaction through sufficiently precise motion, pose, contact, force, and release behavior. It covers physical engagement, grasping, lifting, retention, transport, release, state-changing motion, and post-interaction stability once the relevant target and interaction region are available. Action answers "how does the policy produce the required physical change?"

These definitions form a compact operational decomposition of embodied task execution. The policy's intended target is latent: target selection and motion are behavioral proxies, rather than direct observations of internal intent. The dimensions are functionally distinct: a policy can understand the requested object but bind it to the wrong instance, bind the correct instance but miss the interaction region, or localize the target while failing to generate a stable physical result. The posterior retains these possibilities jointly instead of forcing each failed rollout into one exclusive label.

A.2 INTENTION EVIDENCE

Intention evidence describes what the policy appears to be attempting at a particular interaction stage. It is computed from observable behavior and task metadata; it does not require access to the policy's internal state. Here, "intention evidence names this observable proxy and does not claim to recover internal intention directly. The stage target provides the reference against which the behavior is interpreted.

The current protocol contains the following intention-oriented signals.

- **Target selection (`target_match`, mapped to o_1).** The evaluator identifies the entity or region actually approached, contacted, closed on, or released toward and compares it with the stage target. For grasp stages this is typically the nearest object at the selected gripper-close event. For place stages it is inferred from the release position, target-region

hit, or an available structured relation. The binary signal uses a Bernoulli likelihood with Planning and Grounding as parent dimensions; fractional lane summaries use the weighted form described in Section B.1.

- **Wrong-target context relevance (`wrong_target_context_relevance`, mapped to o_2).** When target selection is wrong, this signal measures whether the selected entity is semantically related to the instruction or task context. The manual likelihood assigns greater Planning-failure support to a high contextual-relevance score for an incorrect target; this direction is a modeling assumption rather than a universally identifiable cause.
- **Wrong-target visual similarity (`wrong_target_visual_similarity`, mapped to o_3).** When target selection is wrong, this signal uses the available lexical and shape-based descriptors to approximate similarity between the selected and intended entities. The field name is retained for compatibility; lexical similarity is a proxy and is not itself a visual measurement. A high score supplies evidence about Grounding under the specified likelihood.
- **Localization error (`localization_error`, mapped to o_{4a}).** After target selection is correct, this continuous signal measures the normalized distance between the interaction point and the intended target point or region. Smaller values indicate closer localization. It is a Grounding signal.
- **Motion-direction match (`motion_direction_match`, mapped to o_7).** For push, pull, press, slide, rotate, and related operations, this continuous score compares the observed motion direction or axis with the task-specified direction or axis. It describes whether the policy is attempting the required operation and is associated with Planning and Grounding in the observation specification.

The same signal name is retained across stages, while the implementation stores each stage instance separately (for example, `o1__grasp_source_object`). This stage-scoped representation avoids multiplying a task-level average together with the stage signals that produced it. The task-level value is retained as a summary for reporting.

A.3 Outcome Evidence

Outcome evidence describes the physical consequence of the attempted interaction. It is evaluated only when the corresponding interaction has occurred and the result is physically assessable. The implemented outcome signals are:

- **Contact or grasp success (`contact_success`, mapped to o_4).** This records whether the policy established valid physical engagement after the target was selected. Depending on the benchmark, the evidence comes from explicit grasp/contact fields or a contact history. Its parent dimensions are Grounding and Action.
- **Lift success (`lift_success`, mapped to o_5).** This records whether the selected object exceeded the stage's lift threshold. In the current protocol it is a binary outcome. A continuous lift-height score is retained as a documented extension point but is not enabled in the current Bayesian parameter table.
- **Object retention (`object_kept`, mapped to o_6).** This records whether the object remained held during transfer without dropping, sliding out, or losing the required physical relation. It is evaluated only after successful contact and lift.
- **Final spatial relation (`final_relation_success`, mapped to o_{13}).** This records whether the required relation, such as `on`, `in`, or `above`, holds after release or at the terminal state. It is a Bernoulli signal with Grounding and Action parents in the observation specification and parameter table.
- **State change (`state_change_success`, mapped to o_{11}).** This records whether a joint, switch, object state, or other task state reaches the expected condition. Examples include an opened drawer, a moved slider, a changed light state, or a target object reaching the required configuration. It is modeled as a Bernoulli signal with Grounding and Action parents.
- **Post-interaction stability (`stability_success`, mapped to o_{12}).** This records whether the achieved state remains stable after the interaction, including the absence of rolling, sliding out, dropping, rebound, or stack collapse. It is associated with Action.

Table 7: Implemented evidence slots and their diagnostic parent dimensions.

Evidence	Signal	Parents	Interpretation
Target selection	o_1	P, G	Whether the selected entity or region matches the stage target.
Wrong-target context relevance	o_2	P	Contextual plausibility of an incorrect target.
Wrong-target visual similarity	o_3	G	Similarity between an incorrect and intended target.
Pick localization	o_{4a}	G	Normalized distance from interaction point to intended target.
Contact or grasp	o_4	G, A	Whether valid physical engagement was established.
Lift	o_5	A	Whether the object exceeded the stage lift threshold.
Retention	o_6	A	Whether the object remained held during transfer.
Motion direction	o_7	P, G	Agreement between observed and expected operation direction.
State change	o_{11}	G, A	Whether the expected joint, switch, or object state was reached.
Stability	o_{12}	A	Whether the achieved state remained stable.
Final relation	o_{13}	G, A	Whether the required final spatial relation was satisfied.

- **Place-target relation (`place_target_relation`).** The implementation also retains a categorical relation diagnostic such as `in`, `out_of`, `on`, or `above`. This diagnostic is useful for reporting the physical outcome, but it is not directly multiplied into the current Bayesian likelihood.

The observation protocol therefore separates evidence about target engagement from evidence about its physical outcome. For example, selecting the correct object and then failing to lift it supplies a different evidence pattern from stably lifting an incorrect object. The distinction is preserved through stage-specific target checks and downstream gating.

A.4 Evidence-to-Dimension Mapping

Table 7 summarizes the current dependency structure. A parent set indicates which latent functional state indexes the corresponding likelihood; it does not assert that the evidence is caused by only one capability in the physical world.

The dependency structure follows a fixed diagnostic order. The evaluator first checks whether the stage target matches. If the target is wrong, contextual relevance and visual similarity help separate Planning and Grounding. If the target is correct, localization, contact, and physical outcomes provide Grounding and Action evidence. A failed downstream outcome is not evaluated as an independent Action failure when the prerequisite target or contact event did not occur.

Some evidence has a benchmark-specific physical interpretation. For example, a place stage may use a release position and a final `on`/`in` relation, while a drawer or slider stage uses contact with a handle, motion along a joint-constrained direction, and a joint-state change. These interpretations share the same capability mapping but require benchmark adapters to export the relevant fields.

B Probabilistic Attribution Details

B.1 Evidence Likelihoods

For each stage-scoped signal whose event gate is active and whose value is available, inference evaluates a likelihood conditional on its parent dimensions. Binary signals use Bernoulli likelihoods. Let $e \in \{0, 1\}$ and let θ_r be the probability of a positive signal for parent-state configuration r. Then

$$p(e \mid r) = \theta_r^e (1 - \theta_r)^{1-e}. \tag{8}$$

Table 8: Bernoulli parameters in the default attribution configuration.

Signal	11	10	01	00
o_1 target match (P, G)	0.90	0.35	0.20	0.08
o_4 contact (G, A)	0.90	0.35	0.55	0.10
o_7 motion direction (P, G)	0.90	0.35	0.20	0.08
o_{11} state change (G, A)	0.92	0.35	0.55	0.12
o_{13} final relation (G, A)	0.92	0.35	0.55	0.12

Continuous signals are first normalized to the unit interval. The current implementation clips values only for numerical evaluation at the open interval boundaries and then uses a Beta density:

$$p(e \mid r) = \mathrm{Beta}(e; \alpha_r, \beta_r) = \frac{e^{\alpha_r - 1}(1-e)^{\beta_r - 1}}{B(\alpha_r, \beta_r)}. \tag{9}$$

The localization error uses a distance convention in which smaller values indicate better localization. Direction-match and diagnostic similarity signals use the convention defined by their evidence extractor; their parameterization is recorded with the corresponding parent state.

Likelihood contributions are accumulated in log space. For a fractional binary summary $\tilde{e} \in [0, 1]$, such as the mean of two arm-specific outcomes, the reported implementation uses

$$\log \widetilde{\ell}(\tilde{e} \mid r) = \tilde{e} \log \theta_r + (1 - \tilde{e}) \log(1 - \theta_r). \tag{10}$$

This is a weighted Bernoulli likelihood contribution, not a Bernoulli probability mass for a fractional event. If $\tilde{e} = (e_1 + e_2)/2$, it equals the average of the two binary log likelihoods; treating both lanes as independent observations would instead give their sum. The two choices assign different evidence strength and must not be interchanged when reproducing a profile. Inference with such summaries is interpreted through the corresponding weighted likelihood factors.

B.2 Parameterization Used by the Current Implementation

The reported default parameter set is the manually specified `stage_plan_manual_v1` configuration in `vla_failure_attribution/bayesian.py`. The prior is uniform over the eight latent states:

$$\pi(z) = \frac{1}{8}.$$

The Bernoulli tables are listed in Table 8. Here, "11", "10", "01", and "00" index (Z_P, Z_G) for o_1 and o_7, and (Z_G, Z_A) for o_4, o_{11}, and o_{13}; a single bit indexes Z_A for o_5, o_6, and o_{12}. Within a selected configuration, the same parameter table is reused for stage-scoped signals; these parameters are manually specified and are not fitted to each evaluated rollout or benchmark during attribution.

For Action-only signals, the values are

signal	$Z_A = 1$	$Z_A = 0$
o_5 lift	0.85	0.15
o_6 retention	0.90	0.25
o_{12} stability	0.65	0.35.

The Beta parameters in the default configuration are

signal	parent state 1	parent state 0
o_2 context relevance	$\mathrm{Beta}(2, 6)$	$\mathrm{Beta}(6, 2)$
o_3 visual similarity	$\mathrm{Beta}(2, 6)$	$\mathrm{Beta}(6, 2)$
o_{4a} localization error	$\mathrm{Beta}(2, 8)$	$\mathrm{Beta}(6, 2)$.

For o_7, the current parameter table contains

(Z_P, Z_G)	11	10	01	00
o_7 continuous Beta form	$\mathrm{Beta}(6, 2)$	$\mathrm{Beta}(2.5, 4.5)$	$\mathrm{Beta}(2.5, 4.5)$	$\mathrm{Beta}(2, 6)$.

The current implementation selects the Bernoulli or Beta form for o_7 from the signal's recorded distribution. Thus, a signal emitted as a continuous direction score uses the Beta table, while a binary signal with the same semantic slot can use the Bernoulli table.

Table 9: Joint-state coding: zero denotes a failure in that dimension.

State	Dimensions in the failure state
000	Planning, Grounding, Action
001	Planning, Grounding
010	Planning, Action
011	Planning
100	Grounding, Action
101	Grounding
110	Action
111	None of the three dimensions

These manually specified likelihoods express diagnostic assumptions. Posterior probabilities are therefore model-based attributions; probabilistic normalization alone does not establish empirical calibration. No EM or other parameter-learning procedure is part of the reported inference.

Two benchmark-specific configurations are present. For RoboTwin, the active `robotwin_legacy_o1` set keeps a separate copy of the default parameters and preserves the target-match table for the registered legacy comparison. For VLA-Arena, the active `vla_arena_stage_plan_manual_v2` set replaces the o_2 and o_3 Beta densities with weak Bernoulli tables,

$$p(o_2 = 1 \mid Z_P = 1) = p(o_3 = 1 \mid Z_G = 1) = 0.30,$$
$$p(o_2 = 1 \mid Z_P = 0) = p(o_3 = 1 \mid Z_G = 0) = 0.70.$$

to match the binary/contextual diagnostics exported by that adapter. The distribution choice follows the signal semantics, rather than using a continuous density for a binary event. The reported $\pi_{0.5}$ VLA-Arena analysis uses this stage-scoped v2 protocol. Consequently, the interface and PGA definitions are shared across benchmarks, but the likelihood configuration is not identical everywhere. Cross-benchmark differences in posterior magnitude may reflect both behavior and observation-model differences.

B.3 Joint Prior, Posterior, and Marginal Attribution

The latent state space contains all eight configurations

$$\mathcal{Z} = \{000, 001, 010, 011, 100, 101, 110, 111\},$$

where the three bits are ordered as (Z_P, Z_G, Z_A). Table 9 makes this convention explicit.

For stage-scoped evidence E, let $\mathcal{I}(E) = \{(k, j) : R_{k,j} = 1\}$ denote the active evidence items. The gate combines physical applicability and observation availability; the implementation's `exists` flag must reflect this inclusion decision. We condition on the resulting gate mask without adding a separate likelihood for the pattern of missing or inapplicable evidence. Assuming conditional independence of the included observations given the joint state, the factorized likelihood is

$$p(E \mid Z = z) = \prod_{i \in \mathcal{I}(E)} p(e_i \mid z_{\mathrm{pa}(i)}). \tag{11}$$

With prior $\pi(z)$, the posterior is

$$p(Z = z \mid E) = \frac{\pi(z) p(E \mid z)}{\sum_{z' \in \mathcal{Z}} \pi(z') p(E \mid z')}. \tag{12}$$

The implementation computes this expression through log likelihoods and log-sum-exp normalization. It reports the complete joint posterior, the most probable joint state

$$\hat{z}_{\mathrm{MAP}} = \arg\max_{z \in \mathcal{Z}} p(Z = z \mid E),$$

and the marginal probabilities of absence of failure

$$p_c(E) = p(Z_c = 1 \mid E).$$

The reported failure attribution is the marginal failure probability

$$q_c(E) = 1 - p_c(E) = p(Z_c = 0 \mid E), \qquad c \in \{P, G, A\}. \tag{13}$$

The three values are overlapping marginal probabilities. They are not mutually exclusive class frequencies and therefore need not sum to one.

For a set of failed rollouts $\mathcal{F}$, the benchmark-level profile used in the paper is the arithmetic mean of the per-rollout failure probabilities:

$$\bar{q}_c = \frac{1}{|\mathcal{F}|} \sum_{\tau \in \mathcal{F}} q_c(E(\tau)). \tag{14}$$

Inference is performed per evaluation unit, and the resulting marginals are then averaged. CALVIN subtask analyses use failed attempted subtasks as the units; episode analyses use failed episodes. No single posterior is inferred by pooling all trajectories into one evidence set. A mean profile cannot establish whether different dimensions co-occur within individual episodes; that requires the per-episode joint posteriors.

The largest mean marginal defines the dominant failure dimension in a condition. It differs from both the most frequent per-episode MAP state and the mode of the mean joint posterior. Because profiles are conditioned on failure, a decrease in a marginal does not by itself establish improved absolute capability. If no failed units are present, the failure-conditioned mean is undefined. If an individual unit has no active evidence, inference returns the prior (0.5 per dimension under the uniform prior), not evidence that a failure is absent.

B.4 Conditional Evidence Availability and Gating

Event gates determine which evidence enters the likelihood. A missing signal means that the corresponding physical event or required field was unavailable or not evaluable; it is not a negative outcome. Such a signal contributes no likelihood factor. Each observation stores an explicit `exists` flag, status, source, reason, and evidence references in the `ObservationValue` record.

The record distinguishes three cases: an observed negative outcome has an active gate and value zero; a missing observation lacks a required measurement; and a not-applicable outcome lacks the prerequisite interaction. Only the first contributes a negative likelihood term. The reported gating rules are:

1. **Target gating.** If a stage target does not match the task-specified target, localization and downstream physical outcomes for that target are marked `not_applicable`. An unrelated collision or motion after an incorrect target is not treated as evidence about successful execution of the intended target.
2. **Grasp/lift gating.** A lift stage requires a matching preceding grasp stage. If the grasp/contact prerequisite is absent or unsuccessful, lift and retention are marked `not_applicable`. Retention is evaluated only after successful lift.
3. **Place-attempt gating.** A place stage is evaluated only when a release, a terminal relation, or another observable place attempt is present. A failed rollout that never reaches the place interaction does not create standalone place-failure evidence.
4. **Place-outcome gating.** Final relation and post-place stability are included only when the relevant pick and place targets are identified and the place outcome is physically evaluable.
5. **Duplicate-contact gating.** For an operate stage that reuses exactly the same first-contact snapshot as a preceding contact stage, target-match and localization signals are marked `not_applicable` for the later stage, preventing the same event from being counted twice.

The stage planner still emits downstream stages for auditability, but gating keeps their unavailable outcomes out of the likelihood. This distinction is particularly important for Action: when the intended target was wrong, later behavior occurs on a different physical path and does not measure the Action quality of the intended interaction. The same rule prevents an unattempted place stage from being interpreted as an additional failure.

The current signal conversion preserves both stage-scoped signals and a task-level arithmetic summary. The reported conversion path prefers stage-scoped entries and falls back to legacy aggregate entries when scoped entries are absent. In the gated protocol, an explicit but inactive scoped record

must not be replaced by an aggregate value that bypasses its gate. Consequently, a summary mean is not multiplied in addition to the stage evidence from which it was computed.

B.5 Worked Example: Gating a Failed Grasp

The real-robot case provides a numerical example of the default model. Use target match $o_1 = 1$, normalized localization error $o_{4a} = 0.13957/0.2 = 0.69785$, and contact success $o_4 = 0$. For $z = (z_P, z_G, z_A)$, the unnormalized weight is

$$w(z) = \frac{1}{8}\,\theta_1(z_P, z_G)\ \mathrm{Beta}(0.69785; \alpha_{4a}(z_G), \beta_{4a}(z_G))\,[1 - \theta_4(z_G, z_A)]\,. \tag{15}$$

Normalizing these eight weights and marginalizing gives

$$q(E) = (0.1860,\ 0.9922,\ 0.6682), \qquad \hat{z}_{\mathrm{MAP}} = 100.$$

The failed contact is observed and diagnostic. Lift and retention are inapplicable because their prerequisites were not satisfied, so neither contributes a likelihood term.

For comparison, incorrectly multiplying an additional lift-failure factor $p(o_5 = 0 \mid Z_A)$ yields $(0.1860, 0.9905, 0.9194)$. This is a calculation illustrating the effect of violating the gate, not an additional measured result. In particular, it spuriously increases Action attribution by treating an unattempted downstream outcome as an observed failure.

B.6 Implementation Notes and Scope

The canonical evidence container is the versioned `EvidenceFrame`. Benchmark-specific planners compile a task into `Stage` records, and benchmark adapters normalize native evaluator outputs into task metadata, robot/action traces, interaction events, scene/object states, relations, and provenance. Shared observation extraction produces `ObservationValue` records and maps them to the Bayesian signal names listed above. The documented extraction entry point is `compute_evidence_observations`; attribution uses the `compute_attribution` path in `vla_failure_attribution/bayesian.py`.

The current method implementation supports the unified observation interface for the benchmark/task protocols registered in the project. A signal may be missing for a particular primitive when the required physical field is not exported by that evaluator. Missing coverage is recorded in the output manifest and does not constitute a failure observation. Details of benchmark coverage, model settings, and the full result tables are reserved for the experimental appendix sections.

C Experimental Setup and Implementation

C.1 Models

The main comparison includes six policies in two groups. Qwen3VL-4B-GR00T, Qwen3VL-4B-PI, and Qwen3VL-4B-OFT share a Qwen3-VL-4B backbone and the reported training setting without VLA pretraining; their action architectures differ. We abbreviate them as Qwen-GR00T, Qwen-PI, and Qwen-OFT. The pretrained group comprises GR00T-N1.7, $\pi_{0.5}$, and OpenVLA. A frozen π_0 is used additionally in the SimplerEnv intervention study. Missing main-table entries are unreported under the current comparison protocol and do not imply zero success.

C.2 Benchmarks and Task Protocols

SimplerEnv. The SimplerEnv Bridge setting contains four single-object manipulation tasks: `PutCarrotOnPlateInScene-v0`, `PutSpoonOnTableClothInScene-v0`, `PutEggplantInBasketScene-v0`, and `StackGreenCubeOnYellowCube BakedTexInScene-v0`. The common protocol decomposes each rollout into a grasp/lift chain followed by a place stage. The supplied four-task records use 400 rollouts per model, with 100 rollouts per task. This count is not assigned to other result rows unless their evaluation manifests confirm the same sampling protocol. The stage evidence covers target selection, localization, contact, lift, retention, place-target intention, final relation, and post-place stability.

Table 10: Representative tasks and stage plans in the evaluated benchmark protocols. The terminal criterion is the native task or simulator success condition exported to the evidence record.

Benchmark	Representative tasks	Task goal or state change	Stage decomposition	Coverage in reported study
SimplerEnv	carrot to plate; spoon to cloth; eggplant to basket; green-cube stack	Pick one source object and place it `on` a plate, cloth, or cube, or `in` a basket; stacking also requires post-place stability.	grasp/approach → lift/keep → place → relation and stability	Four tasks, 100 rollouts per task and model.
CALVIN ABC-D	`lift_*`, `rotate_*`, `push_*`, `move_slider_*`, `open/close_drawer`, `turn_on/off_*`	Change an object, articulated joint, or light state, such as lifting a colored block, moving a slider, opening a drawer, or pressing a switch.	approach/contact plus grasp/lift or operation stages; native oracle checks the state transition	34 atomic task types; sequence and subtask records.
CALVIN ABC-D	`stack_block`, `unstack_block`, `place_in_drawer`, `place_in_slider`, `push_into_drawer`	Establish a final object relation or insertion state, including stable stacking and placement inside an articulated container.	grasp/lift → place or push/insertion; release relation and stability are retained	Included in the same 34-task protocol.
RoboTwin	`pick_diverse_bottles`	Pick two randomly shaped bottles with two arms and lift both above their task thresholds.	two parallel grasp/lift lanes, one per arm	Supplied task-specific records: 100 episodes per model for Qwen-GR00T and $\pi_{0.5}$.
VLA-Arena L0	cake, donut, kiwi, onion, and tomato to box	Pick the named object and place it in the box under the L0 post-training object distribution.	grasp/lift → place → final `in` relation	Five tasks, 100 episodes per task.
VLA-Arena L1/L2	L1 repeats the task form with shifted instances; L2 uses apple, bagel, broccoli, chiffon cake, and lime	Execute the same pick-and-place instruction under task-level zero-shot shifts relative to L0; L2 uses shifted object categories; its PGA profile is inferred from behavior.	grasp/lift → place with target-match, contact, lift, retention, and final-relation evidence	Five tasks per level, 100 episodes per task.

CALVIN ABC-D. CALVIN provides articulated and multi-stage interactions, including lifting, pushing, pulling, sliding, rotating, pressing, drawer operations, stacking, and placement into articulated containers. The local task protocol currently registers 34 atomic task types. Depending on the primitive, a task is compiled into approach/contact, grasp, lift, place, or operation stages. The native benchmark metric is the ABC-D average completed sequence length; subtask success labels are retained for failure filtering and task-level summaries.

RoboTwin. RoboTwin extends the study to dual-arm interaction. The registered `pick_diverse_bottles` protocol contains two grasp/lift lanes, one per arm, and requires both bottle-lifting conditions. The supplied task-specific records contain 100 rollouts each for Qwen-GR00T and $\pi_{0.5}$. These counts describe those records and do not establish the episode count of every maintable RoboTwin aggregate. The protocol aggregates evidence across the two arm lanes; this must retain the weighted-likelihood convention in Section B.1.

VLA-Arena UnseenObjects. The VLA-Arena study evaluates the structured L0–L2 `extrapolation_unseen_objects` condition. Both evaluated policies have 500 episodes per level. The supplied $\pi_{0.5}$ task listing contains five tasks per level and 100 episodes per task, for 1,500 episodes in total. L1 and L2 are task-level zero-shot generalization conditions relative to the L0 post-training data. The condition name describes the distribution shift; it does not uniquely identify a PGA failure dimension.

C.3 Task Descriptions and Benchmark Coverage

Table 10 makes the evaluated task conditions concrete. The table uses representative task names for compactness. The grouped CALVIN rows cover the 34 registered atomic task types, whereas the VLA-Arena rows show the five task instances used at each level of the reported UnseenObjects batch. The reported results use the task protocols in the table; the benchmark name alone is not treated as a capability label.

Supplementary settings. The ManiSkill25 comparison evaluates $\pi_{0.5}$ on post-training and unseen tasks. The recorded DROID case evaluates the feasibility of constructing evidence from physical robot data. Their result scope is kept separate from the main six-policy benchmark matrix.

The task descriptions also clarify why the observation interface is stage based. SimplerEnv and the VLA-Arena rows are dominated by pick-and-place evidence, while CALVIN adds state-changing contact, direction, joint, and insertion stages. RoboTwin keeps the two arms as separate lanes before aggregating their evidence. In every case, the native terminal label is paired with the physical events that led to it, so a failed task can retain evidence from successful upstream stages.

C.4 Evaluation Protocol and Aggregation

For each rollout, the benchmark evaluator exports a versioned `EvidenceFrame`. The benchmark planner compiles the instruction into ordered stages, and the shared observation layer computes stage-scoped `ObservationValue` records. A rollout receives the native benchmark success label from its evaluator. For a benchmark with rollout set $\mathcal{T}$, we define the failure set

$$\mathcal{F} = \{\tau \in \mathcal{T} : y(\tau) = 0\},$$

where $y(\tau)$ is the native success label of the evaluation unit. For CALVIN subtask analysis, τ is an attempted subtask, not the entire five-task sequence. Attribution is computed independently for every $\tau \in \mathcal{F}$. Unattempted downstream subtasks are not inserted into the failure set.

The failure-conditioned profile for capability c is

$$\bar{q}^c = \frac{1}{|\mathcal{F}|} \sum_{\tau \in \mathcal{F}} q_c(E(\tau)), \qquad c \in \{P, G, A\}.$$

The three values are marginal failure probabilities from the same coupled posterior and therefore are not mutually exclusive failure-class proportions. Native performance metrics depend on the benchmark: SimplerEnv and RoboTwin report success rate, CALVIN reports average completed sequence length together with subtask labels, and VLA-Arena reports level-specific success rate.

The evidence manifest distinguishes observed, missing, and not-applicable signals. Coverage requires an explicit denominator (for example, observed signals among physically applicable slots), and is separate from the positive-outcome rate among evaluated events. Neither coverage nor task success measures attribution accuracy. Native performance and attribution should be linked by run identifiers and evaluation-unit IDs before counts from separate records are combined.

For a complete CALVIN evaluation with M sequences, let L_i be the number of completed subtasks and A_i the number attempted in sequence i. The sequence metric is $M^{-1} \sum_i L_i$, whereas attempted-subtask success is $\sum_i L_i / \sum_i A_i$. These denominators differ, so the two metrics cannot be substituted for each other or compared across unmatched run subsets.

C.5 Implementation Details and Reproducibility

The current implementation uses the following registered interfaces and settings:

- Evidence schema: `vla_evidence_v3`; task protocol: `task_protocol_v3`.
- Canonical data structures: `EvidenceFrame`, `Task`, `Stage`, `ObservationValue`, and `AttributionResult`.
- Shared extraction entry point: `compute_evidence_observations.py`. Attribution entry point: `vla_failure_attribution/bayesian.py`.
- Default Bayesian configuration: `stage_plan_manual_v1` with a uniform prior over eight states. The corrected VLA-Arena $\pi_{0.5}$ study uses `vla_arena_stage_plan_manual_v2`; RoboTwin uses its registered separate parameter copy.
- Numeric likelihood evaluation uses $\mathrm{EPS} = 10^{-9}$ for probability and Beta-density boundary protection and uses log-sum-exp normalization.
- Lift thresholds are stored in the stage parameters and are consumed from the evaluator output when available. The registered SimplerEnv pick-and-place stages use $0.02\,\mathrm{m}$, while the VLA-Arena object-lift protocol uses $0.03\,\mathrm{m}$. RoboTwin uses the simulator-owned actor geometry to derive its horizontal grasp-distance scale; it does not reuse a fixed distance scale from another simulator.

Table 11: Main-comparison results. Performance is success rate (%) for SimplerEnv and RoboTwin and average completed sequence length for CALVIN. P/G/A are failure-conditioned marginal probabilities (%).

Benchmark	Model	Performance	P	G	A
SimplerEnv	Qwen-GR00T	49.75	19.08	0.08	54.96
	Qwen-PI	60.25	10.26	0.01	47.14
	Qwen-OFT	45.00	13.99	0.02	43.97
	$\pi_{0.5}$	59.52	18.60	36.61	76.82
	GR00T-N1.7	62.50	18.32	32.03	70.27
CALVIN ABC-D	Qwen-GR00T	4.080	17.35	28.38	54.30
	Qwen-PI	2.520	10.66	29.29	71.06
	Qwen-OFT	3.873	21.99	22.96	61.00
	$\pi_{0.5}$	3.761	28.30	22.90	68.30
	GR00T-N1.7	3.371	30.85	17.51	61.75
	OpenVLA	2.681	24.20	28.50	61.70
RoboTwin	Qwen-GR00T	24.40	18.49	73.49	76.59
	Qwen-OFT	88.00	5.37	71.80	96.32
	$\pi_{0.5}$	11.00	18.59	95.55	97.39
	GR00T-N1.7	9.00	18.09	73.97	52.15
	OpenVLA	18.00	18.49	73.82	63.38

- Inference is per rollout. No integrated EM parameter-learning path is used by the reported attribution entry point.

Reproduction requires matching the policy checkpoint, task/evaluation split, rollout IDs, evidence schema, stage planner, gate configuration, parameter set, and inference command. Runtime settings include random seeds, decoding and sampling parameters, action-chunk length, controller frequency, software versions, and hardware. The supplied records do not consolidate all of these fields; values absent from them are not inferred from policy names or benchmark defaults.

D Additional Experimental Results

D.1 Full Model-by-Benchmark Results

Table 11 reproduces the main comparison using native performance metrics and a consistent percentage scale for PGA. Each PGA entry is a mean over failed evaluation units; it is not a percentage of all trials or an exclusive failure-class frequency. Counts from separate task-specific or partial records are not attached to these aggregates without a confirmed run-level correspondence.

Action leads in all reported SimplerEnv and CALVIN profiles, but Grounding leads for GR00T-N1.7 and OpenVLA on RoboTwin. Thus the evidence supports recurring Action attribution together with setting-dependent failure structure, rather than a universal Action-only explanation. SimplerEnv OpenVLA and RoboTwin Qwen-PI are not included in the current main comparison.

D.2 Task- and Primitive-Level Profiles

Table 12 reports the GR00T-N1.7 CALVIN primitive breakdown, including the additional Switch entry supplied in the appendix records. The same policy exhibits different residual profiles across primitives. Lift has both high Action and substantial Planning attribution; Drawer is Grounding-dominant; and Stack has low success with less separation among the three mean marginals.

These are posterior summaries, not independent ground-truth labels for the latent dimensions. High-success primitives can leave few failures for attribution: Drawer, for example, has only 15 failed subtasks. The Switch attribution should likewise be read with its failure count once that count is available. Aggregate PGA values need not identify a single shared cause across all failures of a primitive.

Table 12: GR00T-N1.7 CALVIN primitive results. N/f gives attempted/failed subtasks. SR and PGA marginals are percentages. The Switch count is not supplied; its dash denotes unavailable count metadata.

Primitive	N/f	SR	P	G	A
Lift	892/117	86.9	61.5	4.5	96.1
Rotate	379/24	93.7	35.2	3.7	43.4
Move	470/29	93.8	26.3	24.3	66.2
Place	469/115	75.5	17.5	35.7	70.1
Push	402/134	66.7	24.5	0.7	64.9
Drawer	586/15	97.4	17.3	74.4	24.1
Switch	–	99.5	0.1	33.3	84.6
Stack	181/137	24.3	24.5	24.2	28.2

Table 13: Qwen-PI CALVIN training dynamics. SR is attempted-subtask success (%); P/G/A are failure-conditioned probabilities (%). Asterisks mark reconstructed intermediate SR values.

Checkpoint	Avg. length	Subtask SR	P	G	A
10k	2.875	82.05	17.40	23.61	63.54
20k	3.643	89.14*	21.54	25.54	58.89
30k	3.549	89.11*	17.97	19.43	58.03
40k	3.673	90.07*	20.85	17.34	57.71
50k	3.777	90.77*	24.09	18.48	58.23
60k	3.792	91.07	21.51	14.58	57.00

D.3 Training Dynamics

The Qwen-PI study evaluates six checkpoints with 1,000 sequences each and reports 23,977 attempted subtasks across checkpoints. Table 13 gives sequence performance and failure-conditioned PGA profiles. Intermediate subtask success rates marked with an asterisk were reconstructed from adjacent-checkpoint changes in the supplied records; they are approximate derived values, not separately verified measurements from raw episode counts.

From 10k to 60k, sequence length increases from 2.875 to 3.792 and five-task completion rises from 37.1% to 62.8%. Action remains the largest failure marginal at every checkpoint. Grounding declines overall, while Planning fluctuates. These results describe changing residual failures alongside improved task completion; the changing failure set prevents interpreting each marginal trend as an absolute capability trajectory.

D.4 Distribution-Shift Results

Table 14 gives both policies' complete L0–L2 profiles. Each condition contains 500 episodes; the failure counts follow from the reported success rates. Action is dominant at every level, but the evolution differs across policies. Qwen-GR00T has increasing Action attribution, whereas $\pi_{0.5}$ has larger Planning and Grounding marginals at L2 alongside a smaller Action marginal.

The supplementary ManiSkill25 task-shift comparison for $\pi_{0.5}$ has success rates of 45.0% on post-training tasks and 1.0% on unseen tasks. Grounding attribution increases from 16.64% to 47.71%, while Action remains high (80.84% and 74.59%). These values provide a further example of changing residual profiles. Task lists, sample counts, and the associated inference configuration are needed to reproduce this supplementary comparison.

D.5 Interaction Outcomes and Evidence Availability

Table 15 makes the gate-conditioned event denominators explicit for $\pi_{0.5}$. Target match is positive in 230 of 500 L2 grasp stages and negative in 270; it is not merely "observed" in 230 stages. Contact, lift, and retention outcomes are then assessed among stages satisfying their preceding prerequisites.

These counts describe where the physical interaction chain ceases to satisfy its conditions. They do not assign ground-truth PGA labels: target mismatch can support Planning or Grounding expla-

Table 14: UnseenObjects results. Each row uses 500 episodes; SR and PGA are percentages, with PGA averaged over the failed episodes.

Model	Level	Failures	SR	P	G	A
Qwen-GR00T	L0	184	63.2	18.58	12.08	79.14
	L1	257	48.6	31.08	5.53	84.43
	L2	384	23.2	8.83	11.14	90.12
$\pi_{0.5}$	L0	190	62.0	18.89	4.32	78.94
	L1	262	47.6	20.31	3.97	80.88
	L2	401	19.8	45.54	35.10	59.48

Table 15: Positive interaction outcomes / prerequisite-qualified stages for $\pi_{0.5}$ UnseenObjects. These counts cover the full condition, including successful and failed episodes, rather than only the failure-conditioned attribution population.

Event and conditioning set	L0	L2
Target match / all grasp stages	499/500	230/500
Contact success / matched stages	323/499	114/230
Lift success / contact-success stages	323/323	110/114
Retention success / lifted stages	303/323	98/110

nations, and failed contact can support Grounding or Action explanations. Nor do they measure evidence coverage, which requires counts of available measurements among applicable slots.

E Qualitative Case Studies

The following examples separate trajectory-level evidence from aggregate condition summaries. Numerical attributions are model outputs or the explicit calculation in Section B.5, rather than manually assigned ground-truth labels.

E.1 Planning-dominated failure

Task. SimplerEnv PutSpoonOnTableClothInScene-v0, representative failure episodes in the Qwen-GR00T record.

Rollout. The policy successfully selects and often grasps the spoon, but the release or termination location does not correspond to the towel region.

Physical evidence. The pick target is correct, while the place-stage target match is 0. The wrong place target is frequently the source spoon or the table surface. In the recorded examples (including ep0, ep1, ep3, ep4, and ep5), the wrong-place relation/context diagnostics are high and the lift/contact chain is otherwise available.

Posterior. The representative Planning-dominated episodes have $q_P(E) = p(Z_P = 0 \mid E) \approx 0.988$. Across the reported failed spoon episodes, the mean Planning and Action marginals are 0.4523 and 0.4702, respectively. The selected case-level pattern therefore does not imply that Planning dominates the full task-level mean.

Diagnosis. The terminal failure is consistent with an incorrect place-stage relation or operation choice. The evidence supports Planning as a major contributor, while the aggregate also contains other evidence patterns and uncertain attributions.

E.2 Grounding-dominant real-robot failure

Task and rollout. Recorded DROID episode 917_1553 attempts to pick up a green cube and place it in a bowl. At the first measured gripper closure, the selected entity matches the required target, while the tool-center point is substantially displaced from it.

Physical evidence. The measured target-to-TCP distance is 0.13957 m in the robot base frame, normalized by 0.2 m to 0.69785. Target match is 1 and contact/grasp success is 0. Failed contact makes lift and retention inapplicable. The example uses the three active observations listed above.

Posterior. Under the default tables and stated gates,

$$(q_P, q_G, q_A) = (0.1860, 0.9922, 0.6682).$$

The MAP state is 100, as reproduced in Section B.5. This matches the main-text case: Grounding is dominant, with a coupled Action component.

Interpretation. The selected target and failed physical engagement support a diagnosis involving localization. This is one recorded example, not a real-world attribution-accuracy evaluation. Pose and contact estimation errors remain possible sources of uncertainty in the evidence.

E.3 Evidence for an Action-related failure

Task and rollout. Qwen-GR00T failures on `PutCarrotOnPlateInScene-v0` include attempts that approach the correct carrot but fail to establish or maintain stable transfer.

Physical evidence. Correct target selection followed by failed contact and successful contact followed by failed lift are distinct evidence patterns. In the former, $o_4 = 0$ is included and o_5 is inapplicable. In the latter, $o_4 = 1$ and $o_5 = 0$ are both included, while retention is inapplicable. A raw zero in a downstream field must not override this distinction.

Interpretation. When target and localization evidence support correct grounding, a failed lift or loss during transfer strengthens an Action explanation. Correct target match alone does not exclude Grounding failure, and contact failure can implicate both dimensions. This example illustrates the required evidence pattern without treating unattempted outcomes as independent Action failures.

E.4 Mixed failure structure under unseen objects

Condition. $\pi_{0.5}$ on UnseenObjects L2 provides a condition-level illustration, rather than a single compound trajectory. Of 500 episodes, 270 have a negative grasp-stage target match. Among the 230 matched stages, 114 establish contact, 110 of these lift the object, and 98 of the lifted stages retain it.

Posterior summary. Across 401 failed episodes, the mean profile is $(\bar{q}_P, \bar{q}_G, \bar{q}_A) = (0.4554, 0.3510, 0.5948)$. Action remains dominant, with substantial Planning and Grounding attribution. This summary permits both heterogeneous failures across episodes and overlapping failures within episodes; it does not distinguish the two without the individual joint posteriors.

Interpretation. An unseen-object condition does not determine a unique failure dimension. The physical records include both target mismatch and unsuccessful downstream execution, and the posterior retains competing functional explanations for those observations.

F Attribution-Guided Intervention Details

F.1 Action Candidate Generation

The registered SimplerEnv intervention uses RL^2-VLA on a frozen π_0 policy. The recorded runtime chain is:

$$\text{frozen policy latent} \rightarrow \text{SAFE failure decision} \rightarrow \text{QAM action candidates}$$
$$\rightarrow \text{candidate combination} \rightarrow \text{CoVer selection} \rightarrow \text{executed action chunk}.$$

The intervention operates at action generation and action selection. It does not update the policy parameters or replace the language and visual inputs. The supplied runtime description identifies these components but does not specify all candidate-generation and filtering settings. The intervention can access the frozen policy latent even though the PGA diagnostic framework itself does not require model-internal access. Keeping the input modalities fixed does not keep later observations identical: changed actions can alter the subsequent closed-loop state.

Table 16: Action-level intervention on 100 fixed SimplerEnv π_0 episodes per task. Repair/damage denominators are original failures/ successes. Changes are percentage points.

Task	Original SR	Corrected SR	Repaired	Damaged	Change
Eggplant	92%	96%	8/8	4/92	+4
Spoon	56%	54%	12/44	14/56	−2
Stack Cube	65%	55%	10/35	20/65	−10
Carrot	51%	61%	23/49	13/51	+10

Table 17: CALVIN targeted recovery records.

Model	Repair operator	Cases	Recovered	Scope
Qwen3VL-OFT	donor suffix	19	3	correct target contact; object not kept
Qwen3VL-OFT	vertical under-lift patch	4	2	target retained; lift height insufficient
Qwen3VL-PI	privileged P/G target patch	4	3 official / 4 target-state	upstream target selection

The CALVIN recovery experiments use deterministic local action patches rather than RL^2-VLA. The Action patches replace a post-contact donor suffix or a vertical lift component while preserving the original task, target object, and trajectory prefix. The P/G recovery experiment replaces the pre-grasp target selection and spatial alignment, then applies the same target-agnostic nearest object operation oracle to the control and patched branches.

F.2 Evaluation Protocol

For RL^2-VLA, the original policy is frozen and the same episode set is compared before and after intervention. We report two transition counts: original failures repaired and original successes damaged. For N paired trials, let n_{01} count failure-to-success transitions and n_{10} count success-to-failure transitions. The change is

$$\Delta\mathrm{SR} = \frac{n_{01} - n_{10}}{N}, \qquad \text{repair rate} = \frac{n_{01}}{N_{\text{failure}}}, \qquad \text{damage rate} = \frac{n_{10}}{N_{\text{success}}}. \tag{16}$$

Multiplying $\Delta\mathrm{SR}$ by 100 gives percentage points. The pairing should preserve the episode definition and initial condition; stochastic policy sampling and simulator settings must be specified separately in the run manifest.

For CALVIN recovery, each case is reset to its sequence initial state and all preceding successful subtasks are replayed. The original failed subtask is replayed to verify deterministic failure reproduction. The intervention is then applied at a frozen physical point, and the official CALVIN task oracle is used for the post-intervention label. Donor selection is fixed before executing the repair. These protocols preserve the transition accounting and avoid selecting a repair after observing its outcome.

The recovery studies are small, targeted utility tests: they ask whether a repair directed at a diagnosed interaction can change its outcome. Their selected-case recovery rates are not directly comparable with full-task success gains, and the operators do not isolate latent PGA dimensions in general. In particular, privileged P/G edits manipulate target selection and alignment jointly. They do not provide separate validation of Planning and Grounding or an attribution-accuracy estimate.

F.3 Per-Task Intervention and Recovery Results

Carrot gains 10 percentage points and Stack Cube loses 10 under the same task-level action intervention. The net effect depends on both repair and damage, not only the number of corrected failures. This contrast supplies initial evidence for diagnosis-informed intervention selection, rather than proof that PGA predicts gains across tasks. The action-generation pipeline includes its own runtime selection; the comparison does not establish prospective per-episode selection by the PGA posterior.

The Qwen3VL-OFT donor-suffix batch deterministically reproduced all 19 original failures and recovered 3/19. The under-lift distance patch recovered 2/4 cases by replacing only the positive vertical lift command while keeping the original prefix and target. The Qwen3VL-PI target-selection study recovered the correct target state in 4/4 cases and the official CALVIN task in 3/4; Planning

and Grounding were jointly manipulated and are not separately validated by this operator. These selected-case results illustrate candidate repair directions. Their success depends on the operator and the physical state at which it is applied; larger prospective comparisons are needed to establish systematic gains from diagnosis-guided selection.