PRISM: PRECISION FORCING FOR GENERALIST VISION-LANGUAGE-ACTION POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Coupling a pretrained vision-language model (VLM) with a diffusion or flow-matching action expert has become the mainstream vision-language-action (VLA) paradigm for embodied policies, yet they remain unreliable once a task demands precise manipulation, e.g., submillimeter cable insertion. We attribute this weakness to a training objective that averages flow-matching errors across contexts and flow times, allocating little supervision to the few decisive moments determining task success. Along the generative path, inference integrates a locally supervised velocity field, so equally penalized errors at different flow times can affect the executed action unequally. Consequently, precision-task performance deteriorates as physical tolerance tightens by orders of magnitude. We call this supervision imbalance across contexts and flow times *Flow-Time Starvation*. The remedy is to jointly align supervision allocation and representation learning with context-dependent precision demand, enabling the policy to recognize when greater precision is needed. We instantiate this precision-forcing strategy as Precision-aware Reweighting and Internalization for Skilled Manipulation (PRISM), which scores precision demand offline, strengthens flow supervision where errors are consequential and action signals are visible, and internalizes that demand in the VLM condition representation. Simulation and real-robot experiments show that PRISM preserves or improves general-task performance while raising mean end-to-end success from 38.3% to 86.7% across drive connector insertion, bimanual connector mating, and screw tightening, substantially extending the precision boundary of existing embodied policies without new sensors or observation modalities.

1 INTRODUCTION

Vision-language-action (VLA) policies have emerged as a central approach to general-purpose embodied control, building on a sequence of advances in robot policy learning. Action chunking and iterative generative modeling established temporally coherent continuous control (Zhao et al., 2023; Chi et al., 2025; Team et al., 2024), and large-scale robot data paired with vision-language representations extended this capability to open-ended instruction following (Brohan et al., 2022; 2023; Kim et al., 2024). Coupling a pretrained vision-language model (VLM) with a diffusion or flow-matching action expert has since become a mainstream VLA paradigm (Black et al., 2024; Intelligence et al., 2025; Shukor et al., 2025; Bjorck et al., 2025; Yuan et al., 2026). Despite broad competence acquired through VLM pretraining and robot post-training, these policies still struggle with precise operations such as inserting a connector into a socket or fastening a screw.

We investigate this precision gap by examining whether training penalties reflect the physical consequences of action errors. Routine motions often tolerate centimeter-scale positional deviations, whereas precise interactions can require millimeter or sub-millimeter accuracy. Yet comparable errors in normalized action space receive similar penalties, even though the same physical deviation can be harmless during transport and decisive during alignment. Brief precision-critical interactions can therefore determine task success while receiving a smaller share of trajectory-averaged supervision than the longer routine motions around them (Wen et al., 2021; Ankile et al., 2024). This discrepancy also extends along the generative path, where inference integrates a velocity field learned from local supervision. Consequently, residuals with equal training penalties can have unequal effects on the executed action because their propagation depends on flow time. We formalize

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.

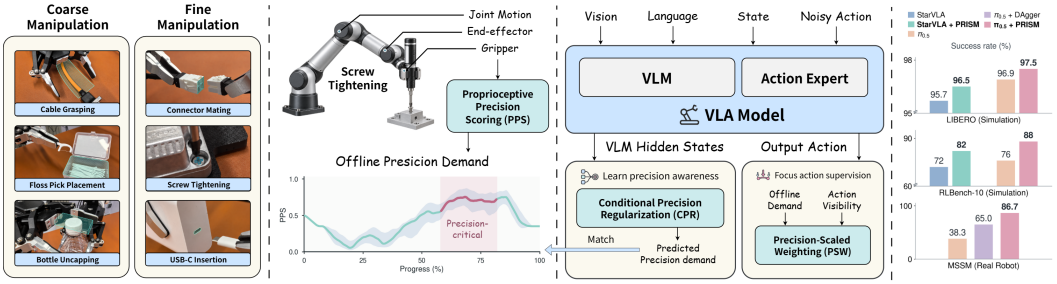


Figure 1: **Overview of PRISM.** PPS estimates precision demand from robot trajectories. CPR learns precision-aware VLM features, while PSW focuses action supervision on precision-demanding contexts with visible action evidence, improving fine manipulation without changing policy inference.

this mismatch between supervision and physical consequence across task contexts and flow times as *Flow-Time Starvation* (see Section 2 for details).

These findings suggest supervising not only the demonstrated action, but also the precision that its execution requires. The resulting training signal should both concentrate learning on consequential errors and make precision demand recognizable from the policy's condition. We implement this *precision-forcing* principle with Precision-aware Reweighting and Internalization for Skilled Manipulation (**PRISM**). Proprioceptive Precision Scoring (PPS) estimates continuous frame-level demand offline from robot trajectories. Precision-Scaled Weighting (PSW) couples this demand with flow-time emphasis and a signal-to-noise ratio (SNR) gate to concentrate additional supervision where action evidence is visible. Conditional Precision Regularization (CPR) makes the same demand predictable from the VLM condition representation. PRISM changes only training, leaving the model architecture and inference procedure unchanged.

Evaluation of PRISM spans standard simulation benchmarks and three extended real-robot tasks with precision-critical interactions. We refer to this setting as Multi-Stage Submillimeter-Tolerance Manipulation (MSSM). The experiments test whether precision forcing extends the precision boundary without sacrificing general breadth. **PRISM** preserves or improves general-task performance and raises mean end-to-end success on MSSM from 38.3% to 86.7%.

Our contributions are summarized below.

- **A formal diagnosis of precision undertraining.** We identify and formalize *Flow-Time Starvation* through an error-propagation bound that accounts for physical tolerance and flow time. Thus, equally penalized velocity errors can affect success differently.
- **A precision-forcing training strategy.** We introduce **PRISM**, where data-driven PPS estimates frame-level precision demand before training, PSW combines this demand with flow-time emphasis and SNR-based visibility, and CPR encourages the VLM condition representation to encode precision demand.
- **Evaluation across breadth and precision.** We evaluate PRISM on standard simulation benchmarks and three MSSM real-robot tasks, demonstrating preserved or improved general-task performance and higher success on precision-critical interactions.

2 PROBLEM FORMULATION

Let $c = (o, l, s)$ denote the current visual observation, language instruction, and proprioceptive state, and let the policy generate an H-step, D-dimensional action chunk $z \in \mathbb{R}^{H \times D}$, flattened as $z \in \mathbb{R}^{HD}$ when convenient. The flow endpoint denotes the clean generated chunk.

Physical precision. Let z^* be the demonstrated chunk, $\hat{z}$ the generated clean chunk, and $e_z = \hat{z} - z^*$ the action prediction error. The map $F_c : \mathbb{R}^{HD} \to \mathbb{R}^m$ describes the physical outcome in m

task-relevant directions. First-order expansion around z^* gives

$$\Delta y_c = F_c(z^* + e_z) - F_c(z^*) \approx J_c e_z, \qquad J_c = \left.\frac{\partial F_c}{\partial z}\right|_{z^*}. \tag{1}$$

If the admissible errors along m task-space directions are $\delta_1(c), \ldots, \delta_m(c)$, define

$$\Lambda(c) = \operatorname{diag}\big(\delta_1(c)^{-2}, \ldots, \delta_m(c)^{-2}\big). \tag{2}$$

The local tolerance-normalized endpoint risk is

$$\mathcal{R}(c) = \mathbb{E}\big[e_z^\top J_c^\top \Lambda(c) J_c e_z \mid c\big]. \tag{3}$$

A tenfold tighter tolerance inflates the squared normalized consequence one hundredfold at unchanged absolute error. The risk also controls task failure. If failure occurs when $\|\Lambda(c)^{1/2} J_c e_z\|_2 > 1$, Markov's inequality applied to the squared norm gives $\Pr(\text{fail} \mid c) \le \mathcal{R}(c)$, so Eq. (3) bounds context-conditional failure probability and formalizes the precision demand of a context.

Local flow objective. We set $\tau = 1$ for noise and $\tau = 0$ for clean actions, so generation integrates from $\tau = 1$ to $\tau = 0$. For $\epsilon \sim \mathcal{N}(0, I)$,

$$x_\tau = (1-\tau)z^* + \tau\epsilon, \qquad u_\tau = \frac{dx_\tau}{d\tau} = \epsilon - z^*. \tag{4}$$

Let v_θ be the learned velocity field and p_τ the flow-time sampling density. The population optimum of squared conditional flow matching is $v^*(x_\tau, \tau, c) = \mathbb{E}[u_\tau \mid x_\tau, \tau, c]$. Along an ideal generation trajectory $x^*(\tau)$ driven by v^*, define the reducible residual $r_c(\tau) = (v_\theta - v^*)(x^*(\tau), \tau, c)$. Its weighted energy over a flow-time region $B \subseteq [0,1]$ is

$$\mathcal{E}_c(B) = \int_B p_\tau(\tau) \|r_c(\tau)\|_2^2 \, d\tau. \tag{5}$$

This residual energy measures the local error penalized by standard flow matching. Its physical consequence depends on propagation to the endpoint.

From residual to endpoint. Linearizing the learned ODE around the ideal trajectory yields

$$\frac{d\,\delta x}{d\tau} \approx G_c(\tau)\delta x + r_c(\tau), \qquad G_c(\tau) = \nabla_x v^*(x^*(\tau), \tau, c). \tag{6}$$

Here δx is the learned trajectory's deviation from x^*, with shared initial noise giving $\delta x(1) = 0$, and G_c is the ideal vector-field Jacobian. The state-transition matrix $\Phi_c(t, s)$ propagates a perturbation from flow time s to t. Residuals in B induce the linearized, tolerance-normalized endpoint perturbation

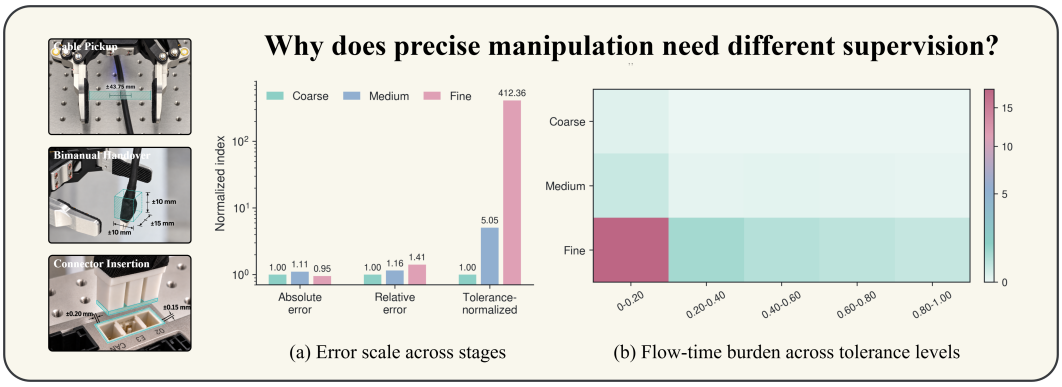
$$\eta_c(B) = -\int_B A_c(\tau) r_c(\tau)\, d\tau, \quad A_c(\tau) = \Lambda(c)^{1/2} J_c \Phi_c(0, \tau). \tag{7}$$

The matrix Φ propagates a local flow error to the clean action, J maps that action error into the physical task space, and Λ measures it relative to what the current context tolerates. Because $\Phi_c(0, \tau)$ depends jointly on flow time and context, and J_c and $\Lambda(c)$ add further context dependence, their product admits no context-independent flow-time weighting in general.

Proposition 1 (Flow-Time Starvation)**.** *For any region B with $p_\tau(\tau) > 0$ and square-integrable residual, the linearized tolerance-normalized endpoint error obeys*

$$\|\eta_c(B)\|_2^2 \le \mathcal{E}_c(B)\, \mathcal{S}_c(B), \quad \mathcal{S}_c(B) = \lambda_{\max}\left[\int_B \frac{A_c(\tau) A_c(\tau)^\top}{p_\tau(\tau)}\, d\tau\right]. \tag{8}$$

Here $\lambda_{\max}$ denotes the largest eigenvalue, and $\mathcal{S}_c(B)$ measures worst-case endpoint sensitivity per unit residual energy. *Derivation.* For any unit task-space direction h, multiplying and dividing the integrand $h^\top A_c(\tau) r_c(\tau)$ by $\sqrt{p_\tau(\tau)}$ and applying weighted Cauchy Schwarz bounds its square by $\mathcal{E}_c(B)$ times $h^\top[\int_B A_c A_c^\top / p_\tau]h$, and maximizing over $\|h\|_2 = 1$ gives Eq. (8) through the Rayleigh quotient. Hence $\mathcal{S}_c(B)$ is the squared operator norm from the weighted local-residual space to the



(a) Error scale across stages (b) Flow-time burden across tolerance levels

Figure 2: **Why precise manipulation needs different supervision.** Left, top to bottom: key frames and tolerances for cable pickup, bimanual handover, and connector insertion, the three stages of the real-robot submillimeter drive-insertion task. (a) Similar action errors incur different tolerance-normalized burdens. (b) Inverse-tolerance-weighted endpoint responses to equal-energy velocity perturbations peak near the clean endpoint for fine manipulation.

tolerance-normalized task space, so one unit of standard FM penalty can leave a large endpoint consequence and the region is starved relative to its physical leverage. The bound is tight for residuals proportional to $p_\tau(\tau)^{-1} A_c(\tau)^\top h_{\max}$, where $h_{\max}$ is a leading eigenvector of the integrated matrix.

Visible supervision. The bound identifies the ideal allocation through Λ, J, Φ, and p_τ, but the first three are unavailable from demonstrations, so a practical objective needs observable surrogates that also avoid amplifying a sample before its action signal becomes visible in the noisy interpolation. For the normalized ground-truth chunk, we therefore define the sample-specific signal-to-noise ratio (SNR) as

$$\sigma_{f,i} = \sqrt{\frac{1}{HD}\sum_{h=1}^{H}\sum_{d=1}^{D}\left(z^*_{i,h,d}\right)^2}, \qquad \mathrm{SNR}_i = \frac{(1-\tau_i)^2\sigma^2_{f,i}}{\tau_i^2 + \varepsilon}. \tag{9}$$

Here $z^*_{i,h,d}$ is the normalized action component at step h and dimension d of sample i, $\sigma_{f,i}$ is its chunk RMS scale, τ_i is the sampled flow time, and $\varepsilon > 0$ stabilizes the denominator.

The gate has a simple probabilistic interpretation. Assume independent action-evidence arrivals with count $N_i \sim \mathrm{Poisson}(\mathrm{SNR}_i\,/\tau_0)$, where $\tau_0 > 0$ controls the visibility saturation scale. Then the probability of observing at least one useful event is

$$\Pr(N_i \geq 1) = 1 - \Pr(N_i = 0) = 1 - \exp(-\,\mathrm{SNR}_i\,/\tau_0). \tag{10}$$

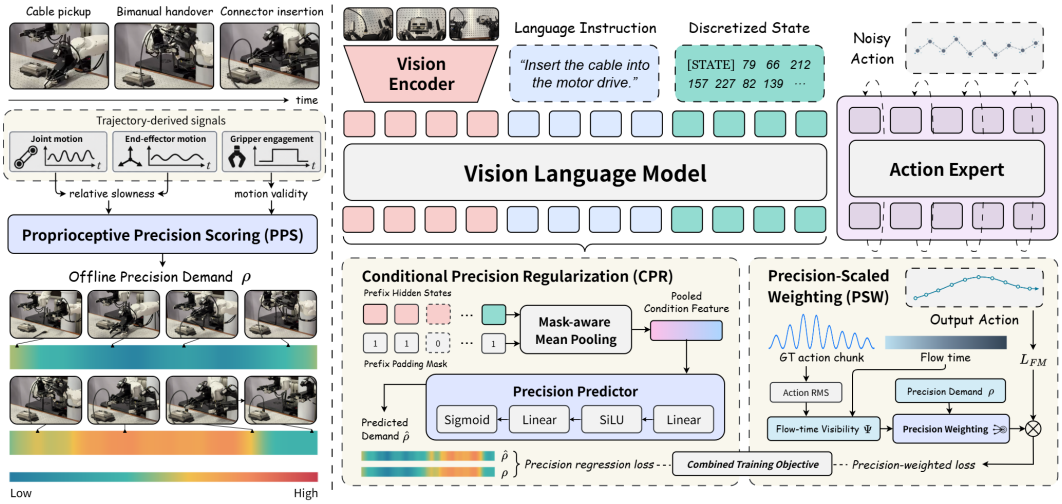
The Poisson assumption converts the SNR into a saturating visibility probability, with Gaussian interpolation setting the SNR and the arrival model gating it.

Empirical diagnosis. Figure 2 diagnoses this mismatch on drive connector insertion, with grasping, handover, and insertion tolerances of 43.75, 10, and 0.15 mm. Absolute FM errors remain comparable, but tolerance-normalized burden rises as tolerance narrows (a). Panel (b) weights endpoint action MSE per unit velocity-perturbation energy by each stage's inverse tolerance relative to grasping. This proxy response peaks for fine manipulation near the clean endpoint (see *Supplementary Material* for the protocol and unweighted responses). These patterns motivate context-dependent precision scoring and clean-end supervision in Section 3.

3 PRISM

3.1 Overview

PRISM uses a shared precision signal to shape both action supervision and the VLM condition representation. PPS computes a frame-level precision-demand label $\rho_i \in [0, 1]$ offline and fixes it



(a) Offline precision estimation (b) Precision forcing during VLA training

Figure 3: **Overview of PRISM.** (a) PPS combines joint motion, end-effector motion, and gripper engagement into offline precision demand. (b) CPR predicts this demand from mask-pooled VLM prefix features, while PSW reweights the flow-matching loss using offline demand and flow-time visibility derived from action RMS and flow time. Both objectives leave inference unchanged.

before optimization. PSW uses this label to allocate more action supervision to precision-demanding samples, while CPR trains the condition representation to encode the same demand. Both objectives use the offline label directly, and the CPR prediction does not feed back into sample weighting. The auxiliary labels and prediction head are discarded after training, leaving the backbone and its standard ODE solver unchanged at inference.

3.2 PROPRIOCEPTIVE PRECISION SCORING

PPS estimates ρ_i from demonstrated joint displacement, end-effector displacement, and gripper state. Because demonstrations record robot motion rather than the latent geometry and contact state that determine physical tolerance, PPS combines motion, engagement, acquisition, and quiescence cues that recur during precision-demanding behavior. For frame i, define the motion measurements and their normalized mixture as follows.

$$\Delta q_i = \|q_i - q_{i-1}\|_2, \quad \Delta e_i = \max_{b\in\{L,R\}} \|e_i^b - e_{i-1}^b\|_2, \quad s_i = \alpha_q \Delta q_i + \alpha_e \Delta e_i. \tag{11}$$

Here q_i and e_i^b are the joint and arm-b end-effector states. The nonnegative mixture coefficients satisfy $\alpha_q + \alpha_e = 1$. Within active frames of the episode, percentile rank converts motion into $\mathrm{slow}_i = 1 - \mathrm{rank}_{\mathrm{episode}}(s_i)$. Let $g_i^{\mathrm{eng}} \in [0,1]$ be normalized gripper engagement and let $t_i \in [0,1]$ combine an acquisition gate, which suppresses rapid closing, and a quiescent-hand gate, which suppresses isolated gripper movement while the arms are static. We set

$$\rho_i = g_i^{\mathrm{eng}} \, \mathrm{slow}_i \, t_i, \tag{12}$$

then apply fixed neutral values to disengaged and near-pause frames, temporal smoothing, and clipping to $[0,1]$ as specified in Section 4.2. Each anchor frame contributes one label to its action chunk, so PSW does not average or maximize future ρ values within the chunk. Section 4.5 evaluates whether this offline behavioral proxy tracks physical precision demand across tasks.

3.3 PRECISION-SCALED WEIGHTING

Section 2 characterizes worst-case endpoint sensitivity through $\mathcal{S}_c(B)$. Its underlying context factor $\Lambda(c)^{1/2} J_c$ and propagation factor $\Phi_c(0,\tau)$ are not directly available during training. PSW approximates these dependencies with a separable surrogate. The precision score ρ_i represents context-dependent demand, a monotone function of τ emphasizes the clean end, and the visibility proba-

bility in Eq. (10) delays amplification until the noisy interpolation contains usable action evidence. Figure 2(b) motivates this separable design through a tolerance-weighted response that combines tighter task constraints with stronger clean-end perturbation effects.

$$\Psi_i = (1-\tau_i)^p \left(1 - e^{-\mathrm{SNR}_i/\tau_0}\right), \qquad w_i = 1 + \gamma \rho_i \Psi_i. \tag{13}$$

The positive hyperparameters p, τ_0, and γ control clean-end emphasis, visibility saturation, and maximum amplification, respectively. Since $\Psi_i \in [0,1]$, the weight satisfies $w_i \in [1, 1+\gamma]$, so PSW only adds supervision to precision-demanding samples and never withdraws supervision from any other sample. The reweighted flow-matching objective for sample i is

$$\mathcal{L}_{\mathrm{PSW},i} = \frac{w_i}{HD} \sum_{h=1}^{H} \sum_{d=1}^{D} \left(v_{\theta,i,h,d} - u_{i,h,d}\right)^2. \tag{14}$$

The weight multiplies the chunk as a whole, so PSW redistributes gradient magnitude across samples and flow times rather than across action dimensions. Flow time alone does not determine action visibility. At the same τ, a small-amplitude precision chunk can have a lower SNR than a larger routine chunk, causing a flow-time-only rule to amplify supervision while the target remains dominated by interpolation noise. The saturating gate delays this amplification until usable action evidence appears, and Section 4.5 evaluates its incremental effect after clean-end weighting. PSW reallocates gradients without changing the flow-time sampler, noise schedule, or number of action experts.

3.4 Conditional Precision Regularization

PSW allocates action supervision according to precision demand, while CPR trains the condition representation to encode that demand. This representation-level supervision helps the policy recognize when greater precision is needed. For the mask-aware pooled VLM condition representation h_{c_i}, a lightweight multilayer perceptron (MLP) head predicts

$$\hat{\rho}(c_i) = \sigma(\mathrm{MLP}(h_{c_i})), \qquad \mathcal{L}_{\mathrm{CPR},i} = (\hat{\rho}(c_i) - \rho_i)^2. \tag{15}$$

Jointly training the head and conditioning backbone allows the precision target to shape the representation itself. PSW continues to use the offline target ρ_i rather than $\hat{\rho}$, keeping sample weighting independent of the model's current prediction. The complete objective reweights the flow-matching loss and adds exactly one auxiliary term,

$$\mathcal{L} = \mathcal{L}_{\mathrm{PSW}} + \lambda_{\mathrm{CPR}} \mathcal{L}_{\mathrm{CPR}}. \tag{16}$$

Beyond the fixed PPS settings, PRISM therefore introduces four scalar hyperparameters, namely p, τ_0, γ, and λ_{CPR}, and no inference-time module.

4 Experiments

4.1 Experimental Setup

Backbones. We evaluate PRISM with $\pi_{0.5}$ and the π-style flow-matching variant of StarVLA (Community, 2026) on simulation benchmarks. The real-robot study uses $\pi_{0.5}$. Each controlled comparison shares the observations, pretrained initialization, demonstrations, update budget, and inference schedule.

Simulation. RLBench-10 evaluates one multi-task policy across ten language-conditioned tasks, while LIBERO contains the Spatial, Object, Goal, and Long suites (James et al., 2020; Liu et al., 2023). We report rollout success by task or suite and its mean. RLBench baselines include OpenVLA, SpatialVLA, π_0, CogACT, CoT-VLA, HybridVLA, and LaST$_0$ (Kim et al., 2024; Qu et al., 2025; Black et al., 2024; Li et al., 2024; Zhao et al., 2025; Liu et al., 2025; 2026). The LIBERO comparison includes SpatialVLA, CoT-VLA, UnifiedVLA, GR00T N1, OpenVLA-OFT, and π_0 (Qu et al., 2025; Zhao et al., 2025; Wang et al., 2026; Bjorck et al., 2025; Kim et al., 2025; Black et al., 2024). Controlled results use each backbone with and without PRISM.

Table 1: RLBench-10 success rates (absolute gains in parentheses). Green highlights PRISM.

Method	Close box	Close laptop	Toilet seat	Sweep	Close fridge	Phone	Umbrella	Frame	Wine	Water plants	Mean
OpenVLA	0.60	0.35	0.75	0.55	0.85	0.20	0.30	0.15	0.20	0.05	0.40
SpatialVLA	0.80	0.70	0.85	0.20	0.80	0.15	0.25	0.40	0.15	0.30	0.46
π_0	0.90	0.60	**1.00**	0.30	0.90	0.25	0.35	0.75	0.05	0.45	0.55
CogACT	0.90	0.80	0.95	0.50	0.85	0.50	0.55	0.45	0.30	0.25	0.61
CoT-VLA	**0.95**	0.75	**1.00**	0.80	0.65	0.50	0.40	0.50	0.55	0.50	0.66
HybridVLA	0.85	0.95	**1.00**	**0.90**	**1.00**	0.50	0.50	0.70	0.50	0.50	0.74
LaST$_0$	**0.95**	0.95	**1.00**	0.80	0.85	0.75	0.75	0.70	0.85	0.60	0.82
StarVLA	0.80	0.95	0.85	0.75	0.90	0.45	0.60	0.60	0.85	0.45	0.72
StarVLA + PRISM	0.85	0.95	0.95	0.85	0.95	0.60	0.70	0.75	**1.00**	0.60	0.82 (+0.10)
$\pi_{0.5}$	0.90	0.95	0.85	0.80	**1.00**	0.55	0.60	0.70	0.80	0.45	0.76
$\pi_{0.5}$ + PRISM	0.90	**1.00**	**1.00**	**0.90**	**1.00**	**0.85**	**0.80**	**0.80**	0.90	**0.65**	**0.88** (+0.12)

Real robot. Multi-Stage Submillimeter-Tolerance Manipulation (MSSM) evaluates extended bimanual tasks with precision-critical alignment and contact stages. Drive connector insertion has a measured geometric clearance of 0.15 mm. We evaluate three MSSM tasks on the Marvin dual-arm robot using 500 demonstrations per task. Each method is evaluated in 20 trials per task. Both end-to-end and individual-stage success rates use a denominator of 20. We report mean completion time over successful trials. A precision stage fails when the robot releases the object or tool after unsuccessful insertion or tightening. Any stage failure makes the full trial unsuccessful. Bimanual mating continues to later connector pairs after an earlier failure, so stage success need not decrease monotonically. We compare $\pi_{0.5}$, its DAgger variant trained with 100 additional corrective demonstrations per task, and PRISM applied to the same base policy.

4.2 IMPLEMENTATION DETAILS

For $\pi_{0.5}$ experiments, we use an action horizon of 15, and simulation datasets use an action stride of 1 at their stored control rate. We optimize $\pi_{0.5}$ for 15,000 steps with AdamW, a global batch size of 192, gradient clipping at 1, an exponential moving average decay of 0.999, and 500 warmup steps followed by a constant learning rate of 5×10^{-5}. Training uses eight NVIDIA H20 GPUs with data parallelism and an FSDP axis of one. The real-robot connector-insertion configuration uses $p = 2$, $\tau_0 = 1$, $\gamma = 3$, and $\lambda_{\text{CPR}} = 0.1$. PPS combines joint and end-effector motion with equal weights, uses a disengaged baseline of 0.35 and an inactive-frame score of 0.5, and applies seven-frame smoothing. Paired baseline and PRISM runs share initialization, data order, update budget, and inference settings. Real-robot experiments use the Marvin dual-arm platform with HIKROBOT MV-CS016-10GC cameras. Demonstrations are collected through teleoperation using a DPVR E4 Black headset and controllers.

Table 2: LIBERO success rates (%) with absolute gains. PRISM models are highlighted in green.

Method	Spatial	Object	Goal	Long	Avg.
SpatialVLA	88.2	89.9	78.6	55.5	78.1
CoT-VLA	87.5	91.6	87.6	69.0	81.1
UnifiedVLA	96.5	96.8	95.6	92.0	95.2
GR00T N1	94.4	97.6	93.0	90.6	93.9
π_0	96.8	98.8	95.8	85.2	94.2
OpenVLA-OFT	97.6	98.4	97.9	94.5	97.1
StarVLA	98.8	**99.6**	95.8	88.4	95.7
StarVLA + PRISM	**99.0**	99.4	97.2	90.2	96.5 (+0.8)
$\pi_{0.5}$	98.8	98.2	98.0	92.4	96.9
$\pi_{0.5}$ + PRISM	98.6	99.4	**98.2**	93.8	**97.5** (+0.6)

4.3 SIMULATION RESULTS

RLBench-10 contains more precision-sensitive interactions, where PRISM raises mean success by 0.10 for StarVLA and 0.12 for $\pi_{0.5}$, demonstrating large gains when fine manipulation determines success. LIBERO is simpler and near saturation, yet PRISM still improves the averages by 0.8 and 0.6 points, respectively, showing that the benefit extends beyond precision-heavy benchmarks. We hypothesize that precision-aware training better distinguishes coarse-motion and fine-adjustment regions of the action space, benefiting general manipulation as well.

4.4 MSSM REAL-ROBOT RESULTS

Across drive connector insertion, bimanual connector mating, and screw tightening (Table 3), PRISM raises end-to-end success from $0.35/0.35/0.45$ to $0.95/0.80/0.85$, versus $0.65/0.60/0.70$ after

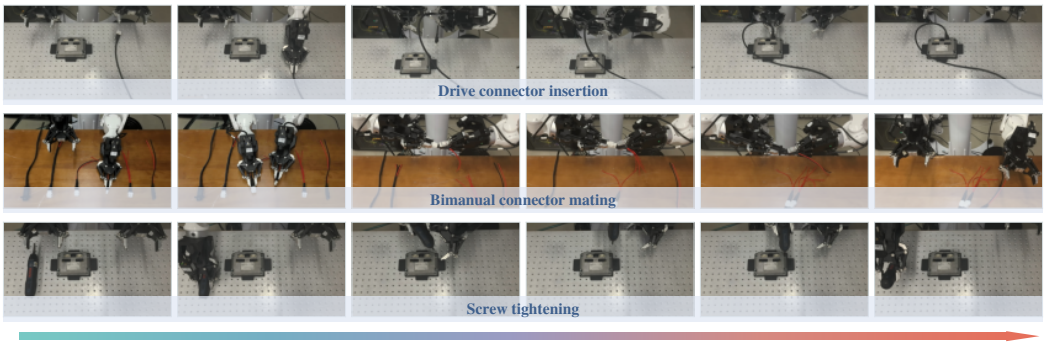


Figure 4: Representative stages of three MSSM tasks, ordered temporally from left to right.

Table 3: MSSM real-robot results. S1–S5 report stage success. SR is end-to-end task success. T is mean completion time in seconds over successful trials. Avg. averages SR over the three tasks.

Method	Drive connector insertion (0.15 mm)					Bimanual connector mating (0.1 mm)							Screw tightening (0.2 mm)					Avg.
	S1	S2	S3	SR↑	$T\downarrow$	S1	S2	S3	S4	S5	SR↑	$T\downarrow$	S1	S2	S3	SR↑	$T\downarrow$	SR↑
$\pi_{0.5}$	1.00	0.90	0.35	0.35	47	0.95	0.95	0.50	0.55	0.65	0.35	65	0.95	0.85	0.45	0.45	37	0.38
$\pi_{0.5}$ + DAgger	1.00	1.00	0.65	0.65	43	0.95	1.00	0.70	0.75	0.85	0.60	59	1.00	0.95	0.70	0.70	35	0.65
$\pi_{0.5}$ + PRISM	1.00	1.00	0.95	**0.95**	**38**	1.00	1.00	0.85	0.90	0.95	**0.80**	**52**	1.00	1.00	0.85	**0.85**	**32**	**0.87**

DAgger. Mean task success increases from 0.38 to 0.87, while successful-trial times decrease from $47/65/37$ to $38/52/32$ s. We observed that DAgger's advantage was less apparent with varied placements or unfavorable pre-alignment configurations. PRISM preserved reliable coarse motions while making more directed local corrections, whereas the baselines more often repeated attempts without resolving misalignment. This suggests improved recovery during precision-critical interaction.

4.5 Ablation Studies

We cumulatively add CPR, precision weighting ($+\rho$), clean-end emphasis ($+\tau$), and SNR gating on drive insertion (Table 4). One attempt is counted each time the robot applies force to insert the connector after an alignment motion. Success rises from 0.35 to 0.45 with CPR and 0.70 with precision weighting, then to 0.85 and 0.95 with clean-end emphasis and SNR gating. We observe more agile fine motions with clean-end emphasis and stronger local adjustment with SNR gating. Attempts decrease from 4.1 to 2.1 across the cumulative design.

4.6 Discussion

PPS estimation accuracy and generality. On 6,692 frames from ten held-out insertion episodes, CPR predicts offline PPS with Spearman/Pearson correlations of $0.915/0.931$ and mean absolute error 0.056 (Figure 5(a)), preserving its continuous demand structure. Across nineteen tasks, PPS highlights fine-operation intervals in mixed coarse–fine trajectories and assigns low demand to coarse-only tasks, supporting cross-task applicability. Task descriptions, PPS computation details, and the corresponding trajectory visualizations are provided in the *Supplementary Material*.

Complementary roles of CPR and PSW. CPR lowers overall residuals but worsens two lower-demand bins, suggesting that recognizing precision demand alone does not ensure balanced action learning. Adding PSW improves every bin (Figure 5(b)). At fixed τ, gains favor higher-SNR groups, with little or negative change at low SNR (Figure 5(c)). These patterns support complementary roles: CPR identifies precision-demanding contexts, while PSW allocates supervision according to both precision demand and action visibility.

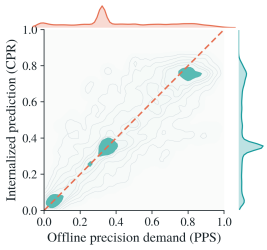
Recognizing precision-critical contexts. Figure 5(d) shows operation-specific CPR sensitivity across four contexts. During handoff, responses concentrate around the held connector and inter-

Table 4: Left-to-right cumulative ablation on drive insertion. Time and attempts average successful trials. +SNR is full PRISM. Arrows indicate better performance.

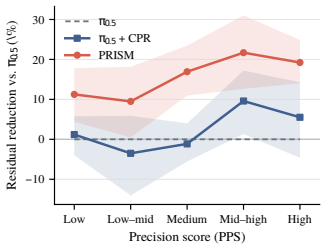
Metric	$\pi_{0.5}$	+ CPR	$+\rho$	$+\tau$	**+ SNR**
Success rate ↑	0.35	0.45	0.70	0.85	**0.95**
Time (s) ↓	47	45	43	40	**38**
Attempts ↓	4.1	3.5	2.7	2.4	**2.1**

Table 5: State-conditioning ablation on drive insertion. Base and PRISM success rates. π_0 retains proprioceptive input to its action expert.

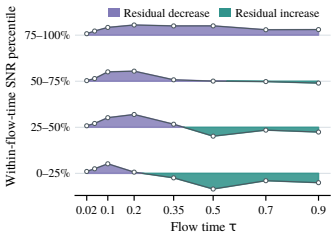
Backbone	VLM state	Base	+PRISM
$\pi_{0.5}$	Yes	0.35	**0.95**
$\pi_{0.5}$	No	0.30	**0.95**
π_0	No	0.25	**0.80**



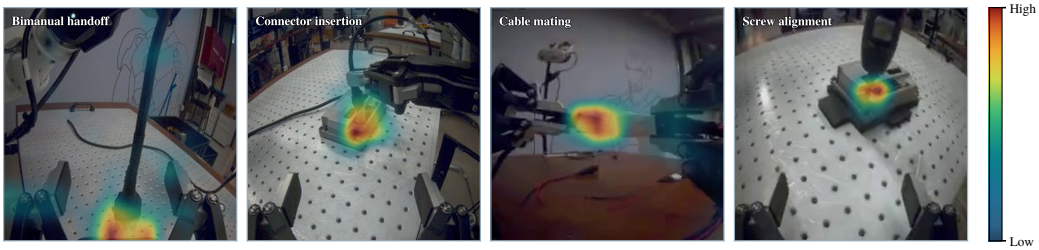
(a) Continuous CPR–PPS agreement



(b) Precision-dependent improvement



(c) Signal-dependent improvement



(d) Visual sensitivity at precision-demanding stages

Figure 5: **Mechanism analysis of precision forcing.** (a) CPR predictions versus offline PPS on held-out insertion trajectories. (b) Residual reduction over $\pi_{0.5}$ across five equal-count PPS bins. (c) Residual profiles in four equal-count SNR groups, formed per flow time within high-PPS frames. Purple/teal indicates a decrease/increase under PSW. (d) Relative CPR-conditioned visual sensitivity across fine-manipulation contexts.

acting grippers. Drive insertion and connector mating emphasize the mating interfaces, while screw tightening elicits responses near the tool–screw contact region. The relevant regions thus vary with the operation, suggesting that the condition representation captures local visual cues for precision demand rather than a fixed scene location.

State-conditioning ablation. We remove state inputs from the $\pi_{0.5}$ VLM during both training and inference (Table 5): baseline insertion success changes from 0.35 to 0.30, while PRISM remains at 0.95. For further validation, we use π_0, which feeds state to the action expert. PRISM raises success from 0.25 to 0.80. Both settings support CPR learning precision demand from visual-language features without requiring VLM state inputs.

5 CONCLUSION

Generalist VLA policies struggle when brief, precision-critical interactions receive insufficient supervision. We address this problem with PRISM, coupling trajectory-derived precision demand with flow-matching loss weighting and precision-aware representation learning. Simulation and real-robot experiments show improved fine manipulation while preserving general-task performance and leaving inference unchanged. A limitation is the two-stage workflow: PPS targets are computed offline before policy training. Learning precision supervision jointly with the policy, without a separate scoring stage, remains future work. We hope PRISM will serve as a solid baseline for extending VLA policies to precision manipulation.

REFERENCES

Lars Ankile, Anthony Simeonov, Idan Shenfeld, and Pulkit Agrawal. Juicer: Data-efficient imitation learning for robotic assembly. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5096–5103. IEEE, 2024.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11462–11471. IEEE, 2022.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7407–7417. IEEE, 2023.

Jialei Huang, Shuo Wang, Fanqi Lin, Yihang Hu, Chuan Wen, and Yang Gao. Tactile-vla: unlocking vision-language-action model's physical knowledge for tactile generalization. *arXiv preprint arXiv:2507.09160*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025.

Zhuoyang Liu, Jiaming Liu, Hao Chen, Jiale Yu, Ziyu Guo, Chengkai Hou, Chenyang Gu, Xiangju Mi, Renrui Zhang, Kun Wu, et al. LaST_0: Latent spatio-temporal chain-of-thought for robotic vision-language-action model. *arXiv preprint arXiv:2601.05248*, 2026.

Xinpan Meng, Siyao Huang, JingPu Yang, Muyuan Ma, Zhenghua Ma, Lijun Han, Gao Yuan, Houcheng Li, and Long Cheng. From reach to insert: Tactile-augmented precision assembly under sub-millimeter tolerances. *arXiv preprint arXiv:2605.04649*, 2026.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 80929–80944, 2026.

Chuan Wen, Jierui Lin, Jianing Qian, Yang Gao, and Dinesh Jayaraman. Keyframe-focused visual imitation learning. *arXiv preprint arXiv:2106.06452*, 2021.

Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. *arXiv preprint arXiv:2503.02881*, 2025.

Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, et al. Forcevla: Enhancing vla models with a force-aware moe for contact-rich manipulation. *Advances in Neural Information Processing Systems*, 38:93409–93439, 2026.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, et al. Qwen-robotmanip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026.

Zongzheng Zhang, Haobo Xu, Zhuo Yang, Chenghao Yue, Zehao Lin, Huan-ang Gao, Ziwei Wang, and Hao Zhao. Ta-vla: Elucidating the design space of torque-aware vision-language-action models. *arXiv preprint arXiv:2509.07962*, 2025.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713. IEEE, 2025.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.