METRICFLY: NUMERICAL INSTRUCTION FOLLOWING VIA ACTION-CONDITIONED CALIBRATION FOR UAV VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Aerial vision-language-action (VLA) policies can generate task-relevant UAV motion, but often fail to meet numerical requirements such as a specified orbit radius or flight altitude. This paper studies how to improve numerical execution by exposing the current metric error to the policy and calibrating the motion it actually generates. We introduce METRICFLY, a two-stage framework that first conditions a π_0 policy on the target, current task metric, and signed residual, then uses a lightweight external calibrator to apply bounded geometric corrections to the sampled action chunk. The calibrator is trained with supervision from an analytic geometric teacher while the fine-tuned policy remains frozen during calibrator training. We evaluate METRICFLY on HUGE-Bench-derived Orbit-R and Orbit-H tasks with 3DGS-rendered visual feedback and externally supplied reference geometry. Explicit numerical conditioning reduces trajectory task-metric MAE by 12.1% and 19.1%, respectively, compared with text-only conditioning. Adding action-conditioned calibration raises full-orbit completion from 2.78% to 93.33% over the metric-conditioned policy alone. Replacing the calibrator's input with a mismatched action sample reduces calibration precision, showing the value of conditioning corrections on the motion being executed. See our Demo for videos.

1 INTRODUCTION

Vision-language-action (VLA) models translate visual observations and language instructions into robot actions (Kim et al., 2025; Black et al., 2024), with recent work extending language-guided control to UAVs (Wang et al., 2025; Guo et al., 2026). Yet executing the right maneuver is not sufficient to follow an instruction with quantitative requirements. A drone instructed to orbit a building at a radius of 25 m and a relative altitude of 40 m must both perform the orbit and regulate its motion to the specified values. We refer to this latter requirement as *numerical regulation*: driving task-relevant physical quantities toward their commanded values and maintaining them during execution. However, our controlled interventions across five VLA policies reveal a failure mode that changing a numerical command can induce negligible motion changes or responses in the wrong direction. These observations expose a gap between producing semantically appropriate actions and controlling the physical quantities specified by an instruction.

Closing this gap requires addressing two distinct problems: how numerical objectives condition action generation, and how generated actions are corrected before execution. First, a commanded value must be interpreted relative to the current physical state: a target radius of 25 m calls for different adjustments when the drone is inside or outside that radius. Prior work on language-model numeracy shows that numerical precision depends on representation and modeling choices (Spithourakis & Riedel, 2018; Wallace et al., 2019), motivating an explicit representation of task targets and their relation to the current state. Second, the required correction depends not only on the state but also on the action proposed by the policy. At a radius of 23 m with a target of 25 m, an outward motion may require different corrections from an inward or tangential motion. A state-and-target-only calibrator does not directly observe which candidate motion it must correct. This distinction motivates explicit numerical conditioning for action generation and action-conditioned calibration for execution.

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.

We introduce METRICFLY, a two-stage framework that implements this separation. In the first stage, we fine-tune a π_0 policy by supplying numerical targets, current task metrics, and signed target-state residuals through its state interface. The language instruction specifies the task semantics, while the controlled numerical values are masked in the text and provided through the structured interface. In the second stage, building on external policy refinement (Yuan et al., 2025), we freeze the fine-tuned policy and train a lightweight calibrator that observes both its decoded action chunk and task-relative geometry. The calibrator predicts radial, tangential, vertical, and yaw corrections, constrained by per-step and chunk-level correction budgets, and applies them before execution. An analytic geometric teacher provides supervision for the calibrator and a reference for evaluating numerical regulation. The first stage thus makes the numerical objective explicit to the policy; the second adjusts the specific motion the policy generates.

We evaluate METRICFLY on controlled Orbit-R and Orbit-H tasks derived from HUGE-Bench (Guo et al., 2026) with 3DGS-rendered visual feedback, kinematic execution, and externally supplied reference geometry. We measure trajectory-level numerical accuracy and full-orbit completion separately, distinguishing regulation error from successful execution of the complete maneuver. Relative to the text-only conditioning, explicit target–state–residual conditioning reduces trajectory task-metric MAE by 12.1% on Orbit-R and 19.1% on Orbit-H. Adding action-conditioned calibration raises full-orbit completion from 2.78% for the metric-conditioned policy alone to 93.33%. Matched–mismatched action interventions further show that calibration precision benefits from observing the particular action chunk being corrected, rather than merely receiving an action input. We additionally evaluate transfer to unseen numerical targets and sensitivity to execution and state perturbations. METRICFLY achieves 100% full-orbit completion at both unseen interpolation target pairs and 88.33%–96.67% across the 11 tested perturbation conditions.

To our knowledge, METRICFLY is *the first study to formulate numerical instruction following for UAV VLAs as an explicit metric-regulation objective and evaluate target-metric accuracy separately from maneuver completion.*

Our contributions are threefold:

- We diagnose numerical-command response failures in UAV VLA policies, identifying negligible and wrong-direction motion responses through controlled interventions that vary commanded values while holding the physical context and task semantics fixed.
- We introduce METRICFLY, a two-stage framework that separates explicit numerical conditioning for VLA action generation from action-conditioned geometric calibration before execution. It combines a structured target–state–residual interface with a calibrator that observes the generated action chunk and predicts corrections subject to per-step and chunk-level budgets.
- We show that calibration precision benefits from observing the specific action chunk being corrected through matched–mismatched action interventions. We further evaluate numerical accuracy and full-orbit completion separately, including transfer to unseen numerical targets and sensitivity to execution and state perturbations.

2 RELATED WORK

Numerical and spatial conditioning for VLAs. VLA policies connect visual observations and language instructions to robot actions through discrete token prediction or continuous action generation (Kim et al., 2025; Black et al., 2024). To strengthen spatial grounding, SpatialVLM and SpatialVLA introduce spatial supervision and representations (Chen et al., 2024; Qu et al., 2025), while VoxPoser and ReKep formulate geometric objectives and constraints (Huang et al., 2023; 2025). Numerical instruction execution further requires relating a commanded quantity to the current physical state. *We address this requirement by separating controlled numerical values from task language and explicitly conditioning the VLA on targets, current metrics, and signed residuals.*

Language-guided UAV control. AerialVLN and TravelUAV study language-guided aerial navigation (Liu et al., 2023; Wang et al., 2025), while AutoFly and AerialVLA explore coarse guidance and interactive dialogue (Sun et al., 2026; Chen et al., 2026). HUGE-Bench includes high-level UAV orbit tasks with instruction-specified radius and altitude (Guo et al., 2026), but its evaluation emphasizes process completion and collision-aware metrics rather than isolating error in regulation

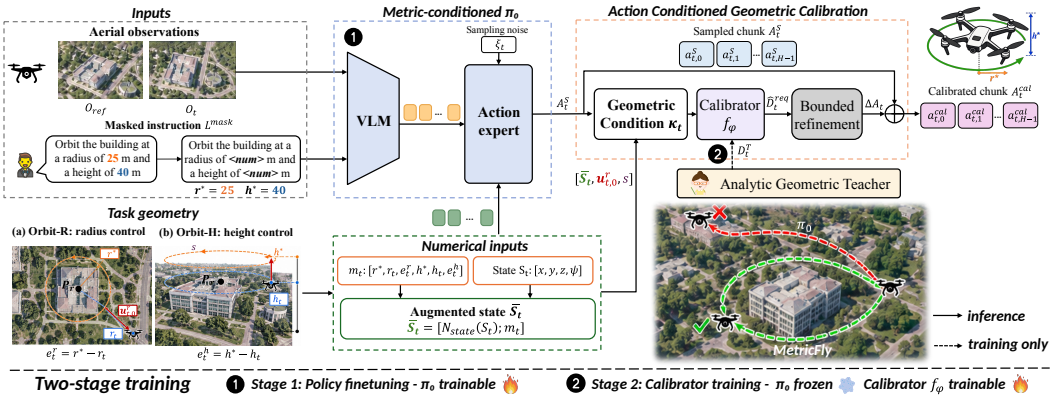


Figure 1: **Overview of METRICFLY.** The π_0 policy π_θ maps images (O^{ref}, O_t), masked language L^{mask}, and augmented state $\bar{\mathbf{S}}_t$ to $\mathbf{A}_t^S$; $\mathbf{m}_t$ encodes targets, current metrics, and signed residuals. From $\mathbf{A}_t^S$ and geometry $\boldsymbol{\kappa}_t$, f_ϕ predicts $\widehat{\mathbf{D}}_t^{\text{req}}$. Budgeted geometric conversion gives $\Delta\mathbf{A}_t$ for execution as $\mathbf{A}_t^{\text{exec}}$. Training fine-tunes then freezes π_θ before fitting f_ϕ to analytic-teacher targets $\mathbf{D}_t^T$. 🔥 and ❄ denote trainable and frozen modules, respectively, at each stage.

of the commanded metric. Such tasks require both executing the requested maneuver and regulating its physical scale. *Our work addresses numerical instruction following for UAV orbit tasks by combining metric-conditioned VLA action generation with external, action-conditioned geometric calibration to regulate orbit radius and flight altitude.*

Action refinement for pretrained policies. Policy Decorator and ResiP learn residual corrections through reinforcement learning while retaining frozen base policies (Yuan et al., 2025; Ankile et al., 2024). An alternative is to select among candidate actions, as in MG-Select and RoboMonkey (Jang et al., 2026; Kwok et al., 2025). For numerical trajectory execution, different sampled actions can require different corrections even under the same state and target. *We adopt supervised residual refinement, using an analytic geometric teacher to train a lightweight calibrator conditioned on the decoded action chunk and task geometry. The calibrator predicts bounded corrections tailored to the sampled motion.*

3 METHODOLOGY

METRICFLY is a π_0-based VLA framework that combines explicit numerical conditioning with external action-conditioned geometric calibration for UAV numerical instruction following. We separate the controlled numerical values from the task language and provide the targets, current task metrics, and signed residuals through the policy's state interface. Conditioned on reference and current images, masked task language, and the augmented state, the VLA generates a continuous action chunk. An external calibrator then uses this sampled chunk and task geometry to predict correction requests, which are bounded and applied before execution. Training proceeds in two stages: we first fine-tune HUGE-Bench π_0, then freeze the entire base policy and train the calibrator using correction targets supplied by an analytic geometric teacher. Figure 1 summarizes the information flow and training stages.

3.1 PROBLEM SETUP AND NUMERICAL INTERFACE

The task takes a reference image O^{ref}, a current image O_t, the UAV state $\mathbf{S}_t$, and an instruction L. The goal is to generate continuous actions that follow the commanded direction, radius, and reference-relative height. Given an externally supplied reference point $\mathbf{P}_r = [x_r, y_r, z_r]^\top$, UAV state $\mathbf{S}_t = [x_t, y_t, z_t, \psi_t]^\top$, and targets $(r^\star, h^\star)$, define

$$\begin{aligned} r_t &= \|\mathbf{S}_{t,xy} - \mathbf{P}_{r,xy}\|_2, \quad e_t^r = r^\star - r_t, \\ h_t &= S_{t,z} - P_{r,z}, \qquad\quad e_t^h = h^\star - h_t, \quad z^\star = P_{r,z} + h^\star. \end{aligned} \tag{1}$$

We decompose the instruction into a masked semantic prompt and numerical targets, $(L^{\text{mask}}, r^\star, h^\star) = \text{Decompose}(L)$. The target–state–residual interface is

$$\mathbf{m}_t = [\widetilde{r}^\star, \widetilde{r}_t, e_t^r/\sigma_r, \widetilde{h}^\star, \widetilde{h}_t, e_t^h/\sigma_h]^\top, \quad \bar{\mathbf{S}}_t = [\mathcal{N}_{\text{state}}(\mathbf{S}_t); \mathbf{m}_t]. \tag{2}$$

Tildes denote normalization with shared training-split means and scales for each metric; residuals use the same scales without mean centering (Appendix B). Reference geometry is externally supplied through these quantities, although the raw reference coordinate is not appended to the policy state.

3.2 Metric-Conditioned Action Generation

Following HUGE-Bench (Guo et al., 2026), the policy uses reference and current images, masked task language, and augmented state to sample $H = 24$ continuous actions. Each decoded action is $\mathbf{a}_{t,h}^S = [\Delta x_{t,h}^S, \Delta y_{t,h}^S, \Delta z_{t,h}^S, \Delta\psi_{t,h}^S]^\top$ in world-frame coordinates. Stacking their transposes gives the sampled chunk $\mathbf{A}_t^S \in \mathbb{R}^{H\times 4}$:

$$\mathbf{A}_t^S = \text{Decode}\big[\pi_\theta(O^{\text{ref}}, O_t, \bar{\mathbf{S}}_t, L^{\text{mask}}; \xi_t)\big]. \tag{3}$$

Here ξ_t is sampling noise. The VLA supplies task-conditioned continuous motion rather than a symbolic maneuver label.

3.3 Action-Conditioned Geometric Calibration

Given the sampled chunk $\mathbf{A}_t^S$ and the direction-conditioned geometry $\boldsymbol{\kappa}_t(s_t)$ defined below, the external calibration stage computes the correction and executed chunk as

$$\Delta\mathbf{A}_t = C_\phi(\boldsymbol{\kappa}_t(s_t), \mathbf{A}_t^S), \qquad \mathbf{A}_t^{\text{exec}} = \mathcal{W}_\psi(\mathbf{A}_t^S + \Delta\mathbf{A}_t). \tag{4}$$

Here $s_t = s_t^T$ during calibrator training and $s_t = \widehat{s}_t$ at learned inference. The composite operator C_ϕ includes geometric processing, MLP prediction, budget clipping, and conversion to world-frame increments specified below. The operator $\mathcal{W}_\psi$ wraps each yaw increment to $(-\pi, \pi]$ and leaves translation unchanged.

Action-derived orbit direction. The tangential correction basis must follow the motion being corrected. At inference, we determine the orbit direction from the complete sampled chunk by a deterministic rule, without reading an expert trajectory label. Let $\mathbf{p}_{t,h}^S$ denote the reference-relative position obtained by rolling out its uncorrected horizontal actions from $\mathbf{S}_{t,xy} - \mathbf{P}_{r,xy}$ after the initialization guard. The radial unit vector $\mathbf{u}_{t,h}^{r,S}$ follows this rollout with the last-valid-axis convention at center crossings; both safeguards are specified in Appendix E.1. Writing $\Pi_{xy}(\mathbf{a}_{t,h}^S)$ for the horizontal displacement, define

$$\gamma_t = \sum_{h=0}^{H-1} \left\langle \Pi_{xy}(\mathbf{a}_{t,h}^S), \begin{bmatrix} -u_{t,h,y}^{r,S} \\ u_{t,h,x}^{r,S} \end{bmatrix} \right\rangle, \qquad \widehat{s}_t = \begin{cases} +1, & \gamma_t \geq 0, \\ -1, & \gamma_t < 0. \end{cases} \tag{5}$$

Positive and negative signs denote counter-clockwise and clockwise motion, respectively. Exactly zero is assigned $+1$; every negative score, however small, is assigned -1. The sign is recomputed for each sampled chunk and enters both the MLP condition and tangential basis. It aligns calibration with the sampled motion, rather than independently enforcing the commanded direction.

Different motions can require different corrections at the same state and target. Let $\mathbf{u}_{t,0}^r$ be the chunk-onset radial axis, obtained from $(\mathbf{S}_{t,xy} - \mathbf{P}_{r,xy})/r_t$ with the same near-zero-radius guard. The calibrator combines the geometric condition with the ordered sampled actions:

$$\begin{aligned} \boldsymbol{\kappa}_t(s) &= \big[\bar{\mathbf{S}}_t; \mathbf{u}_{t,0}^r; s\big], \quad \mathbf{z}_t^S = \text{vec}(\mathbf{A}_t^S) \in \mathbb{R}^{4H}, \\ \widehat{\mathbf{D}}_t^{\text{req}} &= \text{reshape}_{H\times 4}\, f_\phi\big(\text{Std}_{\text{cal}}[\boldsymbol{\kappa}_t(s_t); \mathbf{z}_t^S]\big). \end{aligned} \tag{6}$$

The geometric condition has 13 coordinates. Together with the complete ordered $H \times 4$ action sequence, it gives $13 + 4H = 109$ inputs at $H = 24$. The shared MLP predicts an $H \times 4$ array of radial, tangential, vertical, and yaw correction requests, not a single-axis residual. Continuous inputs use training-split standardization; the direction sign remains discrete. Appendix E specifies the network architecture.

Bounded geometric corrections. Each channel $c \in \{r, \tau, z, \psi\}$ has its own step bound and cumulative absolute budget. Sequential clipping maps each request $\widehat{d}^{c,\mathrm{req}}_{t,h}$ to an applied correction $d^c_{t,h}$, enforcing $|d^c_{t,h}| \le b^{\mathrm{step}}_c$ and $\sum_{h=0}^{H-1} |d^c_{t,h}| \le b^{\mathrm{chunk}}_c$, not a shared four-channel norm constraint. At the corrected reference-relative horizontal position $\mathbf{p}_{t,h}$, let $\mathbf{u}^r_{t,h} = \mathbf{p}_{t,h}/\|\mathbf{p}_{t,h}\|_2$, with the same center-crossing fallback. The world-frame correction is

$$\begin{aligned} \mathbf{u}^\tau_{t,h} &= s_t[-u^r_{t,h,y}, u^r_{t,h,x}]^\top, \quad \mathbf{a}^{\mathrm{exec}}_{t,h} = \mathcal{W}_\psi(\mathbf{a}^S_{t,h} + \Delta\mathbf{a}_{t,h}), \\ \Delta\mathbf{a}_{t,h} &= \left[(d^r_{t,h}\mathbf{u}^r_{t,h} + d^\tau_{t,h}\mathbf{u}^\tau_{t,h})^\top, d^z_{t,h}, d^\psi_{t,h}\right]^\top. \end{aligned} \tag{7}$$

The rows of $\Delta\mathbf{A}_t$ in Eq. equation 4 are $\Delta\mathbf{a}^\top_{t,h}$. The basis follows the corrected rollout, whereas s_t remains fixed within the chunk. Appendix E.1 gives the transition, budget recursion, teacher, and degenerate-radius handling.

3.4 Learning and Information Access

We initialize HUGE-π_0 from the HUGE-Bench checkpoint and fine-tune all base-policy parameters with its original flow-matching objective (Black et al., 2024). The fine-tuned base policy is then frozen before fitting the calibrator. For each sampled chunk, a geometric teacher constructs bounded correction targets $\mathbf{D}^T_t$ under the same budgets and nominal world-frame kinematic transition. It regulates radius and height, activates tangential motion near the target circle, and aligns yaw with the signed tangent, without accessing future expert actions.

During calibrator training, the demonstrated direction s^T_t is used in both $\boldsymbol{\kappa}_t(s^T_t)$ and the signed correction basis. The network is trained with the channel-scaled Huber loss

$$\mathcal{L}_{\mathrm{cal}}(\phi) = \mathbb{E}_{t,\xi_t}\left[\frac{1}{4H}\sum_{h=0}^{H-1}\sum_{c\in\{r,\tau,z,\psi\}} \rho_\delta\left(\frac{\widehat{d}^{c,\mathrm{req}}_{t,h} - d^{c,T}_{t,h}}{s_c}\right)\right]. \tag{8}$$

Here s_c is the scale for correction channel c. At online inference, all learned parameters are frozen and s^T_t is replaced by $\widehat{s}_t$ in both the condition and tangential basis. This training–inference change does not involve retraining or consulting an expert trajectory online. Appendix E.3 gives the base-policy objective and explicit training and inference mappings.

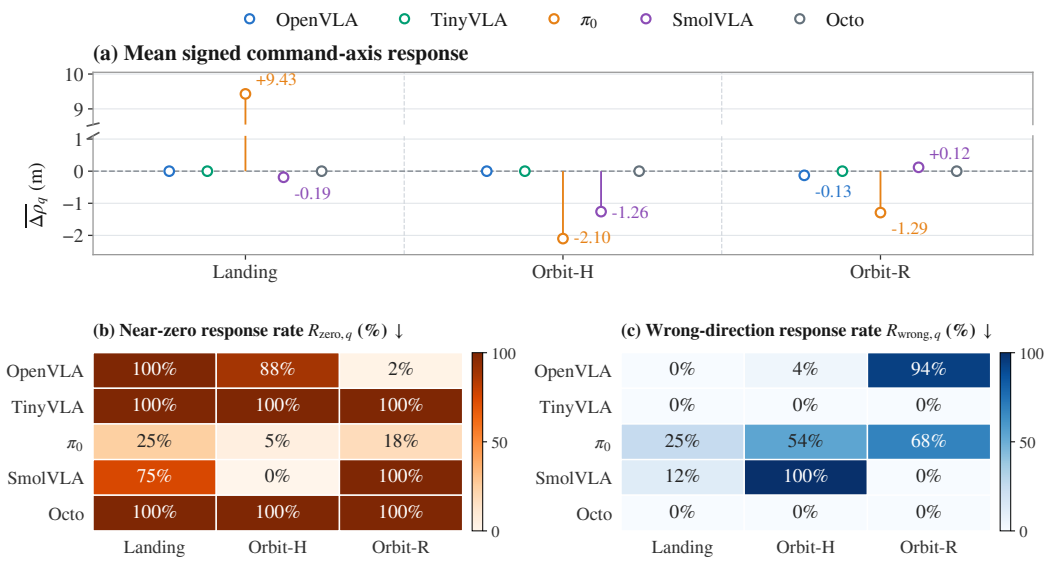
4 Experiments

We organize our experiments around four questions. We first diagnose numerical-command sensitivity in existing VLA policies (RQ1) and examine whether explicit numerical conditioning improves base-action generation (RQ2). We then examine whether calibration benefits from access to the sampled action (RQ3), before evaluating full-trajectory control, generalization to unseen targets, and sensitivity to perturbations (RQ4).

4.1 Experimental Setup

Dataset. We evaluate MetricFly on Metric-State-Hold, a controlled digital-twin dataset constructed from the released HUGE-Bench orbit geometry and 3DGS rendering pipeline (Guo et al., 2026). The dataset contains 2,070 complete trajectories across 23 physical landmarks, with landmark-disjoint training, development, and confirmation splits containing 15, 4, and 4 landmarks, respectively. Orbit-R varies the commanded radius while holding relative height fixed; Orbit-H varies relative height while holding radius fixed. Evaluation uses externally supplied reference geometry and kinematic execution, with 3DGS visual re-observation in the closed-loop experiments; it does not constitute real-flight validation. Dataset construction, collection grids, and split details are provided in Appendix A.

Implementation Details. We initialize HUGE-π_0 from the HUGE-Bench checkpoint and fine-tune all policy parameters for five epochs using its original flow-matching objective; the checkpoint is selected on development data. We then freeze the policy and train the external calibrator for 3,000 updates. Checkpoints and correction bounds are selected using development data before confirmation evaluation. Architecture, normalization, optimization, and implementation details are given in Appendices B and D.1.



	Landing	Orbit-H	Orbit-R
OpenVLA	100%	88%	2%
TinyVLA	100%	100%	100%
π_0	25%	5%	18%
SmolVLA	75%	0%	100%
Octo	100%	100%	100%

	Landing	Orbit-H	Orbit-R
OpenVLA	0%	4%	94%
TinyVLA	0%	0%	0%
π_0	25%	54%	68%
SmolVLA	12%	100%	0%
Octo	0%	0%	0%

Figure 2: **Numerical-command sensitivity across VLA policies.** Panels (a–c) show $\overline{\Delta\rho}_q$, $R_{\text{zero},q}$, and $R_{\text{wrong},q}$ across Landing, Orbit-H, and Orbit-R, with a broken vertical axis in (a).

4.2 RQ1: Numerical-Command Sensitivity

To assess whether numerical changes in an instruction produce appropriate changes in motion, we evaluate OpenVLA (Kim et al., 2025), TinyVLA (Wen et al., 2025), Octo (Octo Model Team et al., 2024), π_0 (Black et al., 2024), and SmolVLA (Shukor et al., 2025) on Landing, Orbit-R, and Orbit-H. Within each matched context, we hold the visual observations, native state, and task semantics fixed and vary only the commanded numerical value. We report mean signed command response $\overline{\Delta\rho}_q$, near-zero rate $R_{\text{zero},q}$, and wrong-direction rate $R_{\text{wrong},q}$. Figure 2 reveals two distinct failure modes: TinyVLA and Octo have $R_{\text{zero},q} = 100\%$ across the tested tasks, whereas π_0 produces negative $\overline{\Delta\rho}_q$ values of -1.29 m on Orbit-R and -2.10 m on Orbit-H. These results show that numerical changes expressed in language do not reliably induce responsive and directionally appropriate motion in the evaluated policies and tasks. They motivate examining whether explicit task-relative numerical inputs can improve action generation. The intervention protocol and response definitions are provided in Appendices F and C.4.

4.3 RQ2: Explicit Numerical Conditioning

To evaluate which numerical inputs improve base-action generation, we compare four π_0 variants with the same backbone, data split, action representation, and optimization protocol. T receives the numerical target through language. For the remaining variants, the controlled numerical value is masked in language while task semantics are preserved: N receives the explicit target, S additionally receives the current task metric, and E further receives the signed target–state residual. We evaluate the variants on matched command triplets using open-loop rollouts of $H = 24$ actions and report equal-weight means over seeds 41–43. We report task-metric mean absolute error (MAE) $E_{\text{metric},q}$, endpoint MAE $E_{\text{end},q}$, expert-referenced 3D trajectory error $E_{\text{traj3D},q}$, strict ordering rate $M_{\text{strict},q}$, and expert-referenced slope error $E_{\text{slope},q}$. The table reports $M_{\text{strict},q}$ as fractions. The input construction, metric definitions, and evaluation protocol are detailed in Appendices B and C.

Table 1 shows that E achieves the lowest $E_{\text{metric},q}$ on both tasks, reducing it relative to T by 12.1% on Orbit-R and 19.1% on Orbit-H. The benefit is not explained by adding more state information alone: S lowers $E_{\text{metric,R}}$ relative to N but increases $E_{\text{metric,H}}$, whereas adding the explicit residual improves both tasks. On Orbit-H, S also improves $M_{\text{strict,H}}$ despite its larger $E_{\text{metric,H}}$, illustrating that correctly ordered responses need not be numerically accurate. These results support explicitly representing the difference between the target and current state during action generation. The

remaining execution errors motivate investigating what information an external calibrator needs to correct the generated motion.

Table 1: **Numerical-state ablation on π_0.** Target–state–residual conditioning yields the lowest $E_{\mathrm{metric},q}$ on both tasks.

Task	Variant	MAE (m) ↓			Numerical value response	
		$E_{\mathrm{metric},q}$	$E_{\mathrm{end},q}$	$E_{\mathrm{traj3D},q}$	$M_{\mathrm{strict},q}$ ↑	$E_{\mathrm{slope},q}$ ↓
Orbit-R	T	2.822	5.116	2.526	0.097	1.015
	N	3.372	5.998	2.874	0.347	1.010
	S	2.607	4.881	2.375	0.458	1.008
	E	**2.481**	**4.468**	**2.107**	**0.514**	**1.002**
Orbit-H	T	3.695	6.151	3.770	0.194	1.001
	N	3.147	5.428	3.524	0.681	0.956
	S	4.477	6.535	4.205	0.764	0.942
	E	**2.989**	**5.395**	**3.087**	**0.944**	**0.888**

4.4 RQ3: Action-Conditioned Calibration

To test whether the sampled action provides useful information beyond the calibration condition, we compare the condition-only calibrator C with the action-conditioned calibrator A. The two methods share the analytic teacher, training chunks, correction bounds, and optimization protocol; A additionally receives the sampled base-action chunk. We evaluate 32 held-out physical contexts per task across three training seeds. At each replanning step, the frozen policy samples an $H = 24$ action chunk, of which the first $K = 6$ calibrated actions are executed. Four replanning cycles, with visual re-observation between cycles, produce a 24-step feedback rollout. Figure 3(b,c) shows that A reduces $E_{\mathrm{traj3D},q}$ relative to C by 0.311 m on Orbit-R and 0.452 m on Orbit-H. Thus, access to the generated motion improves calibration precision beyond the shared condition.

We next test whether this improvement depends on correspondence between the input action representation and the action being corrected. For each intervention, A-Mismatch replaces the calibrator's action input with another sample from the same context–target group, while keeping the checkpoint, calibration condition, base-action chunk being corrected, and correction bounds fixed. Relative to matched conditioning, this substitution increases $E_{\mathrm{traj3D},q}$ by 0.460 m on Orbit-R and 0.644 m on Orbit-H. The mismatch penalty has the same sign in every seed, and the paired 95% cluster-bootstrap intervals exclude zero (Fig. 3(c)). The benefit therefore depends on reading the corresponding action sample, not merely on supplying an additional input vector. The intervention and teacher-correction variation diagnostics are described in Appendices E.4 and H.1.

Action-representation, offline H24, and calibrator capacity and budget ablations are reported in Appendices G.3, I.1, and I.2. We next evaluate full-trajectory performance and generalization.

4.5 RQ4: Full-Trajectory Control and Generalization

Full-trajectory performance. To test whether calibration benefits extend to complete task execution, we evaluate recovery-and-orbit trajectories using the same six-action replanning procedure as RQ3, with matched initial offsets, execution budgets, and stopping rules. The nominal comparison contains 180 rollouts per method across three training seeds. State denotes the frozen metric-conditioned base policy without external calibration. We report joint-entry coverage $M_{\mathrm{rec}}^{\mathrm{joint}}$, conditional post-entry maintenance MAEs $E_{\mathrm{maint},q}$, full-trajectory angular error E_{Θ}, and closure distance D_{close}. Table 2 reports full-lap success rates $M_{\mathrm{lap}}^{\mathrm{joint}}$ of 2.78%, 81.67%, and 93.33% for State, C, and A, respectively. Among the learned calibrators, A achieves the lowest $E_{\mathrm{maint,R}}$ and $E_{\mathrm{maint,H}}$, at 0.053 m and 0.028 m. However, A-Mismatch attains a higher $M_{\mathrm{lap}}^{\mathrm{joint}}$ of 95.56% despite its larger $E_{\mathrm{maint},q}$. These results distinguish improved numerical precision from improved task completion. Representative trajectories are provided in Appendix G (Fig. 5). The full protocol and metric definitions are provided in Appendices G and G.1.

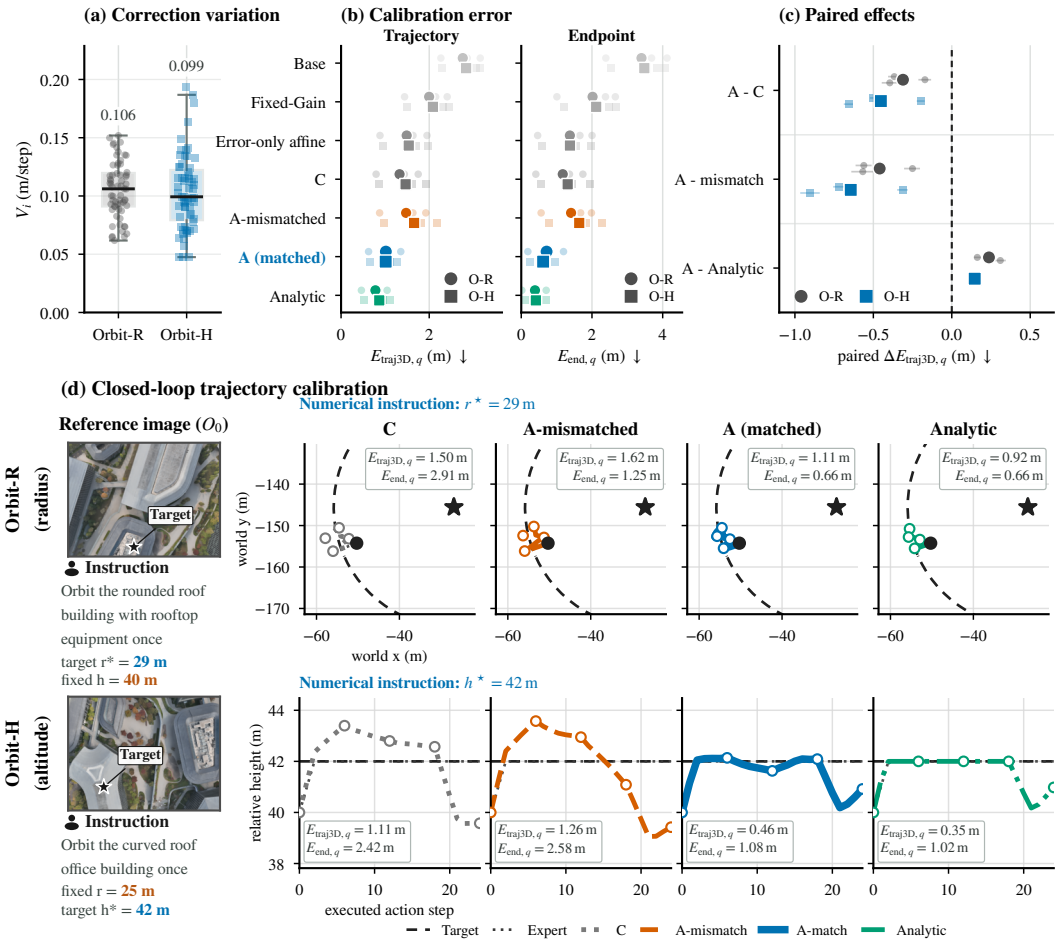


Figure 3: **Short-horizon action-conditioned calibration.** Panels (a–d) show teacher-correction variation V_i, $E_{\text{traj3D},q}$ and $E_{\text{end},q}$, paired differences in $E_{\text{traj3D},q}$, and example rollouts; in (b,c), bars indicate 95% physical-context cluster-bootstrap intervals, and faint and opaque markers denote individual seeds and equal-seed means.

Table 2: **Full-trajectory closed-loop control.** A has the lowest $E_{\text{maint},q}$ among learned calibrators, whereas ANALYTIC-AD has lower $E_{\text{maint},q}$ but lower observed $M_{\text{lap}}^{\text{joint}}$.

Method	Success (%) ↑		$E_{\text{maint},q}$ (m) ↓		Trajectory geometry ↓	
	$M_{\text{rec}}^{\text{joint}}$	$M_{\text{lap}}^{\text{joint}}$	$E_{\text{maint,R}}$	$E_{\text{maint,H}}$	E_{Θ} (rad)	D_{close} (m)
STATE	39.44	2.78	2.860	0.223	4.936	15.742
C	**100.00**	81.67	0.469	0.282	0.960	6.579
A-MISMATCH	**100.00**	**95.56**	0.153	0.032	**0.236**	6.506
A	**100.00**	93.33	**0.053**	**0.028**	0.294	**5.662**
ANALYTIC-AD	100.00	45.56	< 0.001	< 0.001	2.237	19.698
ANALYTIC (oracle)	100.00	100.00	< 0.001	< 0.001	0.162	3.789

Complementary channel-mask ablations show that adding yaw correction improves full-lap success rate $M_{\text{lap}}^{\text{joint}}$, whereas adding the tangential channel does not further improve observed $M_{\text{lap}}^{\text{joint}}$. The masks, complete results, and paired intervals are reported in Appendix G.4.

Information-matched analytic control. To compare learned correction with direct execution of the analytic rule without confounding their direction information, we include ANALYTIC-ACTIONDIR,

abbreviated as ANALYTIC-AD in Table 2. It uses the same ActionDir estimator, geometry inputs, execution protocol, and correction budgets as A. As shown in Table 2, its $E_{\mathrm{maint,R}}$ and $E_{\mathrm{maint,H}}$ are both below 0.001 m, but its full-lap success rate $M_{\mathrm{lap}}^{\mathrm{joint}}$ is 45.56%, compared with 93.33% for A. The learned calibrator therefore has higher observed $M_{\mathrm{lap}}^{\mathrm{joint}}$ but larger $E_{\mathrm{maint},q}$, rather than uniformly outperforming the analytic rule. This comparison shows that lower $E_{\mathrm{maint},q}$ alone does not ensure higher $M_{\mathrm{lap}}^{\mathrm{joint}}$ under the evaluated protocol. The analytic oracle with generator-provided direction is identified separately in the table.

Generalization to unseen numerical targets. To evaluate transfer to numerical commands not seen during training, we freeze the policy and calibration settings and vary either radius or height while holding the other target fixed. The evaluation covers interpolation and extrapolation, with 60 rollouts per method and target pair. Figure 4 shows that A achieves $M_{\mathrm{lap}}^{\mathrm{joint}} = 100\%$ at the two interpolation targets, $(r^\star, h^\star) = (20, 40)$ m and $(25, 34)$ m. The varied-axis errors are $E_{\mathrm{maint,R}} = 0.041$ m for radius interpolation and $E_{\mathrm{maint,H}} = 0.023$ m for height interpolation. The paired comparisons in Appendix G.2 show that A achieves lower $E_{\mathrm{maint},q}$ on the varied axis than C and A-MISMATCH across the tested target settings. Performance is less uniform under extrapolation: $M_{\mathrm{lap}}^{\mathrm{joint}}$ falls to 81.67% at $r^\star = 43$ m, while $E_{\mathrm{maint,R}}$ reaches 0.274 m at $r^\star = 10$ m. The frozen system thus retains a numerical-precision advantage on the tested unseen targets, although $M_{\mathrm{lap}}^{\mathrm{joint}}$ is less consistent under extrapolation. The complete target grid, per-method results, and paired intervals are reported in Appendix G.2.

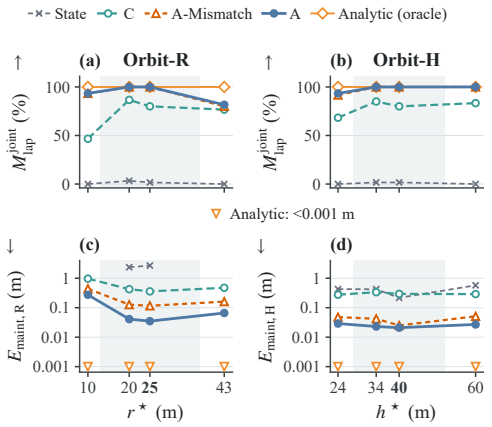


Figure 4: **Generalization to unseen numerical targets.** Panels (a,b) compare $M_{\mathrm{lap}}^{\mathrm{joint}}$ across the five methods. Panels (c,d) compare conditional $E_{\mathrm{maint,R}}$ and $E_{\mathrm{maint,H}}$, respectively, across methods on log scales. Orange downward triangles mark ANALYTIC's reported bounds ($<$ 0.001 m); undefined values are omitted. Shading marks training-target ranges; bold ticks indicate seen targets.

Robustness to execution and state perturbations. To assess sensitivity beyond nominal execution, we evaluate 11 perturbation conditions covering action scaling, one-step delay, persistent local wind bias, state and reference-coordinate noise, actuator limits, yaw–translation coupling, and composite mismatch. Each condition contains 180 rollouts per method. Across the tested conditions, A achieves full-lap success rates $M_{\mathrm{lap}}^{\mathrm{joint}}$ of 88.33%–96.67%, exceeding STATE and C in every condition. Under composite mismatch, $M_{\mathrm{lap}}^{\mathrm{joint}}$ remains 91.11%, while $E_{\mathrm{final,R}}$ and $E_{\mathrm{final,H}}$ increase to 0.490 m and 0.245 m. A-MISMATCH nevertheless has higher observed $M_{\mathrm{lap}}^{\mathrm{joint}}$ in nine of the eleven conditions. The tested conditions and complete result matrix are reported in Appendix I.3.

5 CONCLUSION

We present METRICFLY, a two-stage framework for numerical instruction following in UAV vision-language-action policies. The framework combines explicit target–state–residual conditioning with an external calibrator that predicts bounded geometric corrections from the sampled action chunk and task geometry. Experiments on HUGE-Bench-derived Orbit-R and Orbit-H tasks demonstrate improved numerical accuracy and higher orbit completion rates compared with the uncalibrated base policy. Future work will investigate visually estimated task geometry and validate the framework under physical UAV dynamics.

AI USE STATEMENT

During the preparation of this manuscript, we used OpenAI Codex CLI (version `0.155.1`), powered by GPT-5.6 and accessed during September 2026, as an AI-assisted research and software-engineering tool.

The tool was used in the following parts of the research:

- **Research formulation and experimental design.** Codex was used to help refine the research questions, formulate experimental comparisons, design ablations, specify numerical-control protocols, define evaluation metrics, and organize the train/development/confirmation splits described in the Methodology and Experiments sections.
- **Method and algorithm development.** Codex was used to discuss and draft formulations of the metric-conditioned policy, action-conditioned calibration module, geometric correction rules, correction budgets, and closed-loop evaluation procedures described in the Methodology section.
- **Software implementation.** Codex was used to generate, modify, and debug data-processing scripts, calibration controllers, model-evaluation programs, audit utilities, result-aggregation scripts, and experiment-monitoring utilities used in the code and supplementary artifacts.
- **Experimental execution and result analysis.** Codex was used to launch and monitor author-specified experiments on local and remote GPU servers, inspect execution logs and evaluation receipts, check audit status and artifact integrity, and summarize numerical results produced by the frozen experimental protocols.
- **Manuscript preparation.** Codex was used to draft, translate, restructure, and improve the language of portions of the Methodology, Experiments, Appendix, tables, figure captions, and supplementary documentation.

All AI-assisted code was reviewed by the authors and verified using source inspection, unit tests, integration tests, audit scripts, execution logs, run manifests, artifact checksums, and independent reruns where applicable. The authors checked the consistency of equations, metric definitions, implementation details, experimental protocols, and reported numbers against the source code and raw evaluation artifacts. Numerical results reported in the paper were taken from author-specified evaluation artifacts rather than generated or inferred by the AI tool. Any citations included in the manuscript were independently checked by the authors against the corresponding sources.

The AI tool was not used to fabricate, manually alter, or selectively report data, labels, citations, numerical results, or theoretical claims. It was not given authority to make final decisions about the scientific methodology, experimental inclusion, statistical interpretation, or paper claims. All AI-assisted prose was reviewed and edited by the authors.

The authors take full responsibility for all contents of this paper, including the research design, methodology, implementation, analysis, experimental results, scientific claims, and any text or code produced with AI assistance.

ETHICS STATEMENT

This work evaluates numerical instruction following for UAVs in 3DGS-rendered simulation. Improved numerical accuracy does not guarantee safe real-world flight. Real-world deployment would require further safety validation under actual flight conditions, supported by appropriate safeguards and human oversight. Autonomous UAV capabilities may also be misused for privacy-invasive surveillance. Future applications should respect personal privacy and comply with applicable aviation regulations.

REPRODUCIBILITY STATEMENT

The task construction and split protocol are described in Appendix A; numerical conditioning, replay metrics, statistical procedures, and calibrator details appear in Appendices B– E; supplementary

analyses are in Appendix H. We will release code and configuration files with the final version, subject to the licenses of the underlying benchmark and models.

REFERENCES

Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement—residual RL for precise assembly. *arXiv preprint arXiv:2407.16677*, 2024. URL https://arxiv.org/abs/2407.16677.

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. doi: 10.48550/arXiv.2410.24164. URL https://arxiv.org/abs/2410.24164.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Chen_SpatialVLM_Endowing_Vision-Language_Models_with_Spatial_Reasoning_Capabilities_CVPR_2024_paper.html.

Jinyu Chen, Hongyu Li, Zongheng Tang, Xiaoduo Li, Wenjun Wu, and Si Liu. AerialVLA: A vision-language-action model for aerial navigation with online dialogue. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18161–18169, 2026. doi: 10.1609/aaai.v40i22.38878. URL https://ojs.aaai.org/index.php/AAAI/article/view/38878.

Jingyu Guo, Ziye Chen, Ziwen Li, Zhengqing Gao, Jiaxin Huang, Hanlue Zhang, Fengming Huang, Yu Yao, Tongliang Liu, and Mingming Gong. HUGE-Bench: A benchmark for high-level UAV vision-language-action tasks. *arXiv preprint arXiv:2603.19822*, 2026. doi: 10.48550/arXiv.2603.19822. URL https://arxiv.org/abs/2603.19822.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3d value maps for robotic manipulation with language models. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 540–562. PMLR, 2023. URL https://proceedings.mlr.press/v229/huang23b.html.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4573–4602. PMLR, 2025. URL https://proceedings.mlr.press/v270/huang25g.html.

Subyeok Jang, Dongyoung Kim, Changyeon Kim, Youngsuk Kim, and Jinwoo Shin. Verifier-free test-time sampling for vision-language-action models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=UD4Rw8MOEK.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. 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. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. RoboMonkey: Scaling test-time sampling and verification for vision-language-action models. *arXiv preprint arXiv:2506.17811*, 2025. URL https://arxiv.org/abs/2506.17811.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. AerialVLN: Vision-and-language navigation for UAVs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15384–15394, 2023. doi: 10.1109/ICCV51070.2023.01414. URL https://openaccess.thecvf.com/content/ICCV2023/html/Liu_AerialVLN_Vision-and-Language_Navigation_for_UAVs_ICCV_2023_paper.html.

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. doi: 10.48550/arXiv.2405.12213. URL https://arxiv.org/abs/2405.12213.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Zhao, Jiayuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action model. In *Proceedings of Robotics: Science and Systems XXI*, 2025. doi: 10.15607/RSS.2025.XXI.011. URL https://www.roboticsproceedings.org/rss21/p011.html.

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. doi: 10.48550/arXiv.2506.01844. URL https://arxiv.org/abs/2506.01844.

Georgios P. Spithourakis and Sebastian Riedel. Numeracy for language models: Evaluating and improving their ability to predict numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2104–2115, 2018. doi: 10.18653/v1/P18-1196. URL https://aclanthology.org/P18-1196/.

Xiaolou Sun, Wufei Si, Wenhui Ni, Yuntian Li, Dongming Wu, Fei Xie, Runwei Guan, He-Yang Xu, Henghui Ding, Yuan Wu, Yutao Yue, Yongming Huang, and Hui Xiong. AutoFly: Vision-language-action model for UAV autonomous navigation in the wild. *arXiv preprint arXiv:2602.09657*, 2026. URL https://arxiv.org/abs/2602.09657.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 5307–5315, 2019. doi: 10.18653/v1/D19-1534. URL https://aclanthology.org/D19-1534/.

Xiangyu Wang, Donglin Yang, Hohin Kwan, Jinyu Chen, Hongsheng Li, Yue Liao, Si Liu, Ziqin Wang, and Wenjun Wu. Towards realistic UAV vision-language navigation: Platform, benchmark, and methodology. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Q5eR3F7e90.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. TinyVLA: Toward fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 10(4):3988–3995, 2025. doi: 10.1109/LRA.2025.3547288. URL https://doi.org/10.1109/LRA.2025.3547288.

Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Mengke Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=e5jGTEiJMT.

A METRIC-STATE-HOLD DATASET CONSTRUCTION

A.1 PROVENANCE AND TRAJECTORY COLLECTION

METRIC-STATE-HOLD is a controlled digital-twin extension constructed with the released HUGE-Bench orbit geometry and 3DGS rendering pipeline (Guo et al., 2026); it is neither a real-flight dataset nor an official HUGE-Bench split. We generate 2,070 complete Orbit-R and Orbit-H trajectories from 23 physical landmarks. Each trajectory starts after the approach stage and covers metric recovery, tangent acquisition, one-lap maintenance, and return toward the orbit start. Orbit-R fixes the relative altitude at 40 m, and Orbit-H fixes the radius at 25 m. Each landmark–target pair is initialized at offsets $\{-4, -2, 0, 2, 4\}$ m from the commanded metric.

A.2 COLLECTION GRID

The nominal metric strata specify the physical states around which trajectories are collected; they are not the complete training target support. The full counterfactual and interpolation grids are summarized in Table 3.

Table 3: **Collection grid for Metric-State-Hold.** Counterfactual targets are present in the collected training population; interpolation targets are excluded from policy and calibrator training.

Task	Fixed metric	Nominal strata	Counterfactual target support
Orbit-R	$h = 40$ m	$\{15, 25, 35\}$ m	$\{13, 15, 17, 21, 25, 29, 33, 35, 37\}$ m
Orbit-H	$r = 25$ m	$\{30, 40, 50\}$ m	$\{28, 30, 32, 38, 40, 42, 48, 50, 52\}$ m

Task	Unseen interpolation targets
Orbit-R	$\{20, 30\}$ m
Orbit-H	$\{35, 45\}$ m

For task q, the signed residual is

$$e_{q,k} = y_q^\star - y_{q,k}, \tag{9}$$

where $y_q^\star$ is the commanded metric and $y_{q,k}$ is the current reference-relative metric. Within each counterfactual action-prefix group, we hold fixed the landmark, task reference, observation and UAV-state histories, initial pose, and masked semantic instruction, and vary only $y_q^\star$. Consequently, the group isolates numerical-command effects from changes in the physical context.

A.3 LANDMARK-DISJOINT SPLITS

We partition landmarks by physical identity into training, development, and held-out confirmation splits (Table 4). Every task, target, and initial-offset variant of a landmark remains in the same split. Thus, landmarks, rendered frames, and near-duplicate trajectory prefixes do not cross splits.

Table 4: **Landmark-disjoint data splits for Metric-State-Hold.** The confirmation split is opened only after checkpoint, adapter, and correction-bound selection.

Split	Landmarks	Trajectories	Use
Training	15	1,350	Parameter learning
Development	4	360	Model and bound selection
Confirmation	4	360	Final held-out evaluation
Total	23	2,070	

After registered phase-aware reweighting, the training trajectories yield 10,800 indexed $H = 24$ action-chunk examples. The development split is used exclusively for checkpoint selection, adapter selection, and correction-bound selection. The confirmation split is frozen until all such choices are finalized. The numerical-interpolation evaluation uses the unseen targets in Table 3 and is kept

distinct from the primary RQ3/RQ4 confirmation population. That collection-grid protocol also differs from the additional frozen closed-loop target interventions in Appendix G.2.

A.4 Filtering, deduplication, and weighting

Different command variants from the same physical context share the same landmark and observation/state history. When a downstream component does not receive the command value, these variants are not treated as independent physical samples. We retain at most one such example per physical group or use group-normalized loss weighting so that each physical group has equal total weight. Invalid trajectories, phase registration, rendering parameters, and all filtering thresholds follow the frozen data-generation configuration used for the reported experiments.

B Controlled Numerical Conditioning

B.1 N/S/E interface

The N/S/E variants use the same π_0 architecture, visual observations, masked semantic instruction, native UAV-state pathway, action representation, and flow-matching objective. They differ only in which task-relative quantities populate each task's three numerical-conditioning slots:

$$\begin{aligned}\mathbf{m}^{\mathrm{N}}_{q,k} &= [\widetilde{y}^{\star}_{q}, 0, 0],\\ \mathbf{m}^{\mathrm{S}}_{q,k} &= [\widetilde{y}^{\star}_{q}, \widetilde{y}_{q,k}, 0],\\ \mathbf{m}^{\mathrm{E}}_{q,k} &= [\widetilde{y}^{\star}_{q}, \widetilde{y}_{q,k}, \widetilde{e}_{q,k}].\end{aligned} \tag{10}$$

The final MetricFly base policy uses the E representation. N and S are controlled numerical-interface ablations. For the joint Orbit-R/Orbit-H policy, the task-specific triplets for $q = \mathrm{R}$ and $q = \mathrm{H}$ are concatenated in that order, yielding the six numerical slots in Eq. equation 2. The same N/S/E rule is applied independently to both task metrics, including the metric held fixed by a task's collection protocol.

All variants use the same masked semantic prompt L^{mask}_{q}. Therefore, the comparison isolates explicit task-relative numerical state rather than changing the semantic-language input. The zero entries are constant placeholders used to ablate sample-dependent access to the corresponding metric fields. Because the inputs are normalized, zero numerically coincides with the training-set mean; it therefore removes sample-specific variation while retaining that constant baseline.

B.2 Shared normalization

For each task q, normalization statistics are computed only from the controlled training split. Let

$$\mathcal{Y}^{\mathrm{tr}}_{q} = \{y^{\star}_{q,i}\}_{i \in D_{\mathrm{tr}}} \uplus \{y_{q,i}\}_{i \in D_{\mathrm{tr}}}, \tag{11}$$

where $\uplus$ denotes multiset union. We compute

$$\mu_q = \mathrm{Mean}(\mathcal{Y}^{\mathrm{tr}}_{q}), \qquad \sigma_q = \mathrm{Std}(\mathcal{Y}^{\mathrm{tr}}_{q}), \tag{12}$$

and use

$$\widetilde{y}^{\star}_{q} = \frac{y^{\star}_{q} - \mu_q}{\sigma_q}, \qquad \widetilde{y}_{q,k} = \frac{y_{q,k} - \mu_q}{\sigma_q}, \qquad \widetilde{e}_{q,k} = \frac{e_{q,k}}{\sigma_q}. \tag{13}$$

No separate residual statistics are estimated. Thus, E provides an explicit error coordinate without changing the underlying numerical scale.

B.3 Matched training

N, S, and E are trained as separate policies. Across variants, we keep fixed the policy architecture, initialization protocol, numerical-state token position, training data, number of updates, optimizer, learning-rate schedule, full-parameter policy fine-tuning configuration, and matched random seeds. No variant-specific auxiliary supervision is added to the N/S/E comparison.

C ACTION-CHUNK ROLLOUT EVALUATION

Notation. An evaluation instance is indexed by i and its task by q. For Orbit-R and Orbit-H, $q \in \{\mathrm{R}, \mathrm{H}\}$ and the realized metrics are radius and reference-relative height, respectively. The target is $y^{\star}_{i,q}$, and $y_{i,q,h}$ is the metric after h executed actions. Actions are indexed by $h = 0, \ldots, H-1$; the resulting states are indexed by $h = 1, \ldots, H$, with $h = 0$ denoting the initial state. The evaluated horizon is H; K remains the number of actions executed before replanning, not the number of independent sampled chunks. Scalar task errors are measured in meters and angular errors in radians. Rates are defined on $[0, 1]$ and multiplied by 100 only when displayed as percentages. A sample index is omitted only for a specified single-trajectory example. Unless a protocol states otherwise, metric means are computed within a seed and then summarized as specified in Appendix D. The H24 confirmation metric instead averages per-chunk RMSE within landmarks and weights four landmark means equally (Appendix G.1).

C.1 OPEN-LOOP ACTION-CHUNK ROLLOUT

For evaluation instance i, the policy samples an action chunk from its available observation and state. The evaluator rolls out a fixed prefix under the additive world-frame kinematic transition, without future observations within that prefix. Let $\mathcal{M}_q^{(h)}$ evaluate the task metric after its first h actions:

$$y_{i,q,h} = \mathcal{M}_q^{(h)}\left(\mathbf{A}_i; \mathbf{S}_{i,0}, \mathbf{P}_{r,i}\right). \tag{14}$$

The evaluator computes reference-relative metrics from the resulting state and reference geometry. The error definitions below also apply to fixed-length feedback rollouts, using their executed state sequences rather than a single open-loop chunk.

C.2 PRIMARY TRACKING AND TRAJECTORY METRICS

For evaluated instances $\mathcal{E}_q$, define $N_q = |\mathcal{E}_q|$ and the signed target residual

$$e_{i,q,h} = y^{\star}_{i,q} - y_{i,q,h}. \tag{15}$$

The endpoint and trajectory task-metric mean absolute errors are

$$\begin{aligned} E_{\mathrm{end},q} &= \frac{1}{N_q} \sum_{i \in \mathcal{E}_q} |e_{i,q,H}|, \\ E_{\mathrm{metric},q} &= \frac{1}{N_q} \sum_{i \in \mathcal{E}_q} \frac{1}{H} \sum_{h=1}^{H} |e_{i,q,h}|. \end{aligned} \tag{16}$$

These quantify deviation from the commanded radius or reference-relative height, not from the expert path. Let $\mathbf{x}^{W}_{i,h} \in \mathbb{R}^3$ and $\mathbf{x}^{E,W}_{i,h}$ denote aligned predicted and expert world-frame positions. The expert-referenced 3D trajectory error is

$$E_{\mathrm{traj3D},q} = \frac{1}{N_q} \sum_{i \in \mathcal{E}_q} \frac{1}{H} \sum_{h=1}^{H} \left\| \mathbf{x}^{W}_{i,h} - \mathbf{x}^{E,W}_{i,h} \right\|_2. \tag{17}$$

This is the mean Euclidean position error, referred to as 3D trajectory MAE; it is not a coordinate-wise MAE or an XYZ RMSE. The separate offline H24 representation ablation reports $E^{H24}_{\mathrm{xyzRMSE}}$: a mean of coordinate-wise per-chunk RMSE values with equal landmark weighting. Equations equation 57 and equation 58 specify the order of operations.

We additionally report endpoint success at tolerance $\epsilon_{\mathrm{end}} = 1\,\mathrm{m}$ and reduction in absolute task error:

$$\begin{aligned} M_{\mathrm{end},q} &= \frac{1}{N_q} \sum_{i \in \mathcal{E}_q} \mathbf{1}\left[|e_{i,q,H}| \le \epsilon_{\mathrm{end}}\right], \\ \Delta|e|^{(h)}_{i,q} &= |e_{i,q,0}| - |e_{i,q,h}|. \end{aligned} \tag{18}$$

Positive $\Delta|e|^{(h)}_{i,q}$ indicates progress toward the commanded metric.

C.3 COMMAND-AXIS RESPONSE

Let $\mathcal{P}_q$ contain matched command pairs. Pair i shares the same physical context and uses targets $y^{\star,+}_{i,q} > y^{\star,-}_{i,q}$. The difference in the realized physical metric after h actions is

$$\begin{aligned}\Delta\mathcal{M}^{(h)}_{i,q} = \mathcal{M}^{(h)}_q\left(\mathbf{A}_i(y^{\star,+}_{i,q}); \mathbf{S}_{i,0}, \mathbf{P}_{r,i}\right) \\ - \mathcal{M}^{(h)}_q\left(\mathbf{A}_i(y^{\star,-}_{i,q}); \mathbf{S}_{i,0}, \mathbf{P}_{r,i}\right).\end{aligned} \tag{19}$$

For a pre-specified physical tolerance ϵ_a, define a pair score $\omega^{(h)}_{i,q}$ and its mean:

$$\begin{aligned}\omega^{(h)}_{i,q} &= \begin{cases} 1, & \Delta\mathcal{M}^{(h)}_{i,q} > \epsilon_a, \\ \frac{1}{2}, & |\Delta\mathcal{M}^{(h)}_{i,q}| \le \epsilon_a, \\ 0, & \Delta\mathcal{M}^{(h)}_{i,q} < -\epsilon_a, \end{cases} \\ M^{(h)}_{\mathrm{ord},q} &= \frac{1}{|\mathcal{P}_q|}\sum_{i\in\mathcal{P}_q}\omega^{(h)}_{i,q}.\end{aligned} \tag{20}$$

This tolerance-aware score concerns realized task metrics, not the projected sampled-action response below. It does not determine whether the motion is appropriate relative to the current state. The physical response gain is

$$G^{(h)}_{\mathrm{ord},q} = \frac{1}{|\mathcal{P}_q|}\sum_{i\in\mathcal{P}_q}\frac{\Delta\mathcal{M}^{(h)}_{i,q}}{y^{\star,+}_{i,q} - y^{\star,-}_{i,q}}. \tag{21}$$

Both the rollout horizon h and tolerance are fixed within a comparison.

C.4 MATCHED COMMAND-RESPONSE METRICS

For each matched command pair $i \in \mathcal{P}_q$, let $\mathbf{u}^+_{i,q}$ be the registered positive task axis, held fixed at the initial state. It is the outward radial axis for Orbit-R and the upward axis for Orbit-H; the Landing task uses its registered upward axis. The signed difference between the larger- and smaller-command action chunks is

$$\begin{aligned}\Delta\rho_{i,q} &= \sum_{h=0}^{H-1}\left[\Pi_{xyz}(\mathbf{a}^+_{i,h}) - \Pi_{xyz}(\mathbf{a}^-_{i,h})\right]^\top \mathbf{u}^+_{i,q}, \\ \overline{\Delta\rho}_q &= \frac{1}{|\mathcal{P}_q|}\sum_{i\in\mathcal{P}_q}\Delta\rho_{i,q}.\end{aligned} \tag{22}$$

The positive sign corresponds to the larger command. Summation, not temporal averaging, gives a displacement in meters. At a pre-specified tolerance ϵ_ρ, near-zero and wrong-direction rates are

$$\begin{aligned}R_{\mathrm{zero},q} &= \frac{1}{|\mathcal{P}_q|}\sum_{i\in\mathcal{P}_q}\mathbf{1}[|\Delta\rho_{i,q}| \le \epsilon_\rho], \\ R_{\mathrm{wrong},q} &= \frac{1}{|\mathcal{P}_q|}\sum_{i\in\mathcal{P}_q}\mathbf{1}[\Delta\rho_{i,q} < -\epsilon_\rho].\end{aligned} \tag{23}$$

Low $R_{\mathrm{wrong},q}$ alone does not imply reliable instruction following when $R_{\mathrm{zero},q}$ is high. These evaluation tolerances do not alter the zero-threshold ActionDir decision in Eq. equation 5.

For matched residual triplet $i \in \mathcal{T}_q$, the fixed-axis response is

$$\rho_{i,q}(e) = \sum_{h=0}^{H-1}\Pi_{xyz}(\mathbf{a}_{i,h}(e))^\top \mathbf{u}^+_{i,q}. \tag{24}$$

Write $\rho_{i,q,-} = \rho_{i,q}(-\delta)$, $\rho_{i,q,0} = \rho_{i,q}(0)$, and $\rho_{i,q,+} = \rho_{i,q}(+\delta)$, where $\delta > 0$ is the matched residual magnitude. Superscript E denotes the corresponding expert response. The model and expert response slopes are

$$\begin{aligned}\beta_{i,q} &= \frac{\rho_{i,q,+} - \rho_{i,q,-}}{2\delta}, \\ \beta^E_{i,q} &= \frac{\rho^E_{i,q,+} - \rho^E_{i,q,-}}{2\delta}.\end{aligned} \tag{25}$$

Their discrepancy and the strict triplet-ordering rate are

$$E_{\mathrm{slope},q} = \frac{1}{|\mathcal{T}_q|} \sum_{i \in \mathcal{T}_q} |\beta_{i,q} - \beta^E_{i,q}|,$$
$$M_{\mathrm{strict},q} = \frac{1}{|\mathcal{T}_q|} \sum_{i \in \mathcal{T}_q} \mathbf{1}[\rho_{i,q,+} > \rho_{i,q,0} > \rho_{i,q,-}]\,. \tag{26}$$

Responses and residuals are measured in meters, so slopes and $E_{\mathrm{slope},q}$ are dimensionless. Unlike $M_{\mathrm{ord},q}$, $M_{\mathrm{strict},q}$ uses strict inequalities and gives no half credit for a tie; the two scores must not be interchanged.

C.5 State-relative correction and zero-residual holding

The physical metric change from the initial state is

$$\Delta y^{(h)}_{i,q} = y_{i,q,h} - y_{i,q,0}. \tag{27}$$

Partition the evaluation instances into $\mathcal{E}^+_q$, $\mathcal{E}^-_q$, and $\mathcal{E}^0_q$ according to the sign of the initial residual $e_{i,q,0}$. Let $\mathcal{E}^{\mathrm{nz}}_q = \mathcal{E}^+_q \cup \mathcal{E}^-_q$. For a fixed physical response tolerance τ_q, the correction rate is

$$M_{\mathrm{corr},q} = \frac{1}{|\mathcal{E}^{\mathrm{nz}}_q|} \sum_{i \in \mathcal{E}^{\mathrm{nz}}_q} \mathbf{1}\Big[\mathrm{sgn}(e_{i,q,0})\Delta y^{(h)}_{i,q} > \tau_q\Big]\,. \tag{28}$$

The positive- and negative-axis correction rates are

$$M_{\mathrm{out},q} = \frac{1}{|\mathcal{E}^+_q|} \sum_{i \in \mathcal{E}^+_q} \mathbf{1}\Big[\Delta y^{(h)}_{i,q} > \tau_q\Big]\,,$$
$$M_{\mathrm{in},q} = \frac{1}{|\mathcal{E}^-_q|} \sum_{i \in \mathcal{E}^-_q} \mathbf{1}\Big[-\Delta y^{(h)}_{i,q} > \tau_q\Big]\,. \tag{29}$$

These correspond to outward/inward motion for Orbit-R and upward/downward motion for Orbit-H. For zero-residual instances, direction is undefined, so we report holding drift and the holding rate:

$$D_{\mathrm{hold},q} = \frac{1}{|\mathcal{E}^0_q|} \sum_{i \in \mathcal{E}^0_q} |\Delta y^{(h)}_{i,q}|,$$
$$M_{\mathrm{hold},q} = \frac{1}{|\mathcal{E}^0_q|} \sum_{i \in \mathcal{E}^0_q} \mathbf{1}\Big[|\Delta y^{(h)}_{i,q}| \le \tau_q\Big]\,. \tag{30}$$

Conditional scores with empty evaluation subsets are undefined, not zero.

D Statistical Protocol

Controlled comparisons use three matched training seeds, not independent sample multipliers. For comparisons summarized within each seed, each of $10{,}000$ hierarchical paired bootstrap replications proceeds as follows:

1. resample physical anchor or trajectory blocks within each seed;
2. retain every counterfactual command group within a sampled block;
3. compute the paired method effect within each seed;
4. average the three seed-level paired effects.

We report percentile 95% confidence intervals from that bootstrap distribution, together with individual seed values and their mean and standard deviation. For results reported as equal-seed summaries, the mean is $\overline{m} = B_{\mathrm{seed}}^{-1} \sum_{b=1}^{B_{\mathrm{seed}}} m^{(b)}$, with $B_{\mathrm{seed}} = 3$; conditional subsets are formed within each

seed. This rule does not apply to the H24 confirmation XYZ RMSE: its estimator first averages per-chunk errors within each landmark across the evaluated seeds and then averages the four landmark means equally. Its confidence intervals are computed by resampling those four means, as specified in Appendix G.1. Lower error and drift metrics are better; larger ordered-response, state-correction, holding, and success rates are better.

The correction-channel confirmation comparison is a separate exception: targets, offsets, and seeds are averaged within each physical context before equal weighting of the four contexts. Its 95% exact cluster-bootstrap intervals enumerate all 256 context resamples, rather than using the within-seed bootstrap above (Appendix G.4).

For the N/S/E ablation, we evaluate N–S and S–E paired effects separately. For trajectory calibration, we evaluate A–C and matched–mismatched effects using the same sampled base chunks whenever the comparison permits. We do not interpret command ordering alone as evidence of metric accuracy; endpoint metric error and state-relative correction are reported jointly.

D.1 Additional Implementation Details

Base-policy optimization. We initialize the base policy from the public HUGE-π_0 checkpoint released by HUGE-Bench and fine-tune all base-policy parameters using the original flow-matching objective. We do not introduce auxiliary trajectory, phase-classification, geometric, or action-smoothness losses during this stage. Each training example contains an $H = 24$ expert action chunk in the world-frame action convention $[\Delta x, \Delta y, \Delta z, \Delta\psi]$. The task-relative numerical variables are normalized using statistics computed exclusively from the training split.

The joint Orbit-R and Orbit-H training set contains $10{,}800$ action chunks per epoch, with $5{,}400$ chunks from each task. We train each policy for five complete epochs with a batch size of one, corresponding to $54{,}000$ optimizer updates. Optimization uses AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, and weight decay 10^{-10}. The learning rate is linearly warmed up for the first $1{,}000$ updates to 2.5×10^{-5} and then cosine-decayed to 2.5×10^{-6}. We clip the global gradient norm to 1.0 and do not use an exponential moving average of the model parameters.

Calibrator optimization. The base-policy parameters are frozen before calibrator training. For every sampled base-policy chunk, we construct the calibrator input from the normalized task condition and the complete ordered sequence of $H = 24$ decoded base actions. Each action retains all four coordinates $[\Delta x, \Delta y, \Delta z, \Delta\psi]$, giving a $4H = 96$-dimensional action representation. Continuous calibrator inputs are standardized using training-split statistics only. The calibrator is a two-hidden-layer multilayer perceptron with hidden dimensions $(128, 128)$ and SiLU activations. Its linear output head jointly predicts correction requests for all four geometric channels at all $H = 24$ action steps; budget clipping bounds the applied residuals.

We train the calibrator for $3{,}000$ optimizer updates using AdamW with a learning rate of 10^{-3} and weight decay 10^{-4}. Each task-balanced minibatch contains 64 Orbit-R and 64 Orbit-H examples. The supervision targets are generated by the geometric teacher described in Sec. 3.4, and the prediction loss is the Huber penalty with $\delta = 1$. We evaluate the calibrator on the development split every 500 updates. Neither policy nor calibrator normalization statistics are recomputed on development or confirmation data.

Correction constraints. Requested corrections are projected onto the feasible set before being added to the base action. Metric-aligned translational corrections use a maximum magnitude of $1.0\,\mathrm{m}$ per action step and an accumulated absolute budget of $10\,\mathrm{m}$ per $H = 24$ chunk. Yaw corrections are limited to $0.3\,\mathrm{rad}$ per step and $3.0\,\mathrm{rad}$ per chunk. The chunk-level budget is applied sequentially in temporal order, so later residuals cannot exceed the budget remaining after earlier corrections. The same projection operator and bounds are used when constructing teacher targets and when executing learned corrections.

Checkpoint selection. We save a base-policy checkpoint after every training epoch and select checkpoints using only the development contexts. All runs are trained for the full five-epoch budget before selection. For the three reported random seeds, the selected checkpoints correspond to epoch 4 for seed 33, epoch 3 for seed 34, and epoch 4 for seed 35. Calibrator checkpoints are selected independently using their development evaluations. The confirmation contexts are not used

for checkpoint selection, hyperparameter tuning, input standardization, or correction-bound selection.

Closed-loop inference. At each replanning cycle, the base policy generates an $H = 24$ action chunk using ten flow-decoding steps. The calibrator then predicts and bounds the correction sequence for that sampled chunk. We execute only the first $K = 6$ corrected actions before rendering a new observation, updating the UAV state, recomputing the task-relative numerical condition, and sampling a new chunk. This receding-horizon procedure is repeated until the task-specific completion or stopping condition is reached.

Closed-loop evaluation uses 3DGS-rendered visual re-observation and additive world-frame kinematic replay. It therefore evaluates visual closed-loop control under the decoded action contract, but does not constitute evidence of physical UAV dynamics or real-world flight. For paired comparisons, we hold the initial condition, numerical target, action-sampling seed, replanning interval, correction bounds, and stopping criterion fixed across methods.

Compute. Training and evaluation jobs were scheduled across two servers equipped with NVIDIA A100 GPUs. Independent seeds and evaluation conditions were run in parallel when resources permitted. Unless otherwise stated, results are reported over seeds 33, 34, and 35.

E Trajectory-Calibrator Implementation Details

E.1 Geometric Calibration Details

Coordinates and correction basis. For one chunk, we suppress the replanning index t on per-step quantities. Decoded actions are world-frame increments $\mathbf{a}_h^S = (\Delta x_h^S, \Delta y_h^S, \Delta z_h^S, \Delta\psi_h^S)^\top$. Initialize $\mathbf{p}_0 = \mathbf{S}_{t,xy} - \mathbf{P}_{r,xy}$, $z_0 = S_{t,z}$, and $\psi_0 = \psi_t$. The desired circle has radius $r^\star > 0$ and world-frame height $z^\star = P_{r,z} + h^\star$. At predicted step h, the radial, direction-signed tangential, and vertical axes are

$$\mathbf{u}_h^r = \frac{\mathbf{p}_h}{\|\mathbf{p}_h\|_2}, \quad \mathbf{u}_h^\tau = s_t \begin{bmatrix} -u_{h,y}^r \\ u_{h,x}^r \end{bmatrix}, \quad \mathbf{u}^z = (0,0,1)^\top. \tag{31}$$

The normalized radial expression applies above the threshold specified below; center crossings use the last valid radial axis. The sign is $+1$ for counter-clockwise motion and -1 for clockwise motion. Training and the privileged analytic reference use $s_t = s_t^T$; learned online calibration uses $s_t = \widehat{s}_t$. The horizontal basis follows the corrected predicted position, rather than remaining fixed at chunk onset.

The projections and their zero-padded injections are

$$\begin{aligned} \Pi_{xy}(\mathbf{a}) &= [\Delta x, \Delta y]^\top, & \Pi_{xyz}(\mathbf{a}) &= [\Delta x, \Delta y, \Delta z]^\top, \\ \mathcal{I}_{xy}([v_x, v_y]^\top) &= [v_x, v_y, 0, 0]^\top, & \mathcal{I}_{xyz}([v_x, v_y, v_z]^\top) &= [v_x, v_y, v_z, 0]^\top. \end{aligned} \tag{32}$$

Writing the bounded correction as $\mathbf{d}_h = (d_h^r, d_h^\tau, d_h^z, d_h^\psi)^\top$, we obtain the corrected chunk prediction:

$$\begin{aligned} \Delta\mathbf{s}_{h,xy}^{\mathrm{cal}} &= \Pi_{xy}(\mathbf{a}_h^S) + d_h^r \mathbf{u}_h^r + d_h^\tau \mathbf{u}_h^\tau, \\ \Delta z_h^{\mathrm{cal}} &= \Delta z_h^S + d_h^z, \\ \Delta\psi_h^{\mathrm{cal}} &= \mathrm{wrap}(\Delta\psi_h^S + d_h^\psi), \\ \mathbf{p}_{h+1} &= \mathbf{p}_h + \Delta\mathbf{s}_{h,xy}^{\mathrm{cal}}, \\ z_{h+1} &= z_h + \Delta z_h^{\mathrm{cal}}, \\ \psi_{h+1} &= \mathrm{wrap}(\psi_h + \Delta\psi_h^{\mathrm{cal}}). \end{aligned} \tag{33}$$

Here wrap maps angles to $(-\pi, \pi]$. The main-text operator $\mathcal{W}_\psi$ applies this wrapping to the yaw coordinate of each action and leaves its translational coordinates unchanged.

Independent correction budgets. Each channel $c \in \{r, \tau, z, \psi\}$ has its own step bound b_c^{step} and absolute chunk budget b_c^{chunk}. A requested value $\widehat{d}_h^c$ is applied as

$$\begin{aligned} b_{h,c}^{\text{rem}} &= \min\left(b_c^{\text{step}}, \left[b_c^{\text{chunk}} - \sum_{j<h} |d_j^c|\right]_+\right), \\ d_h^c &= \text{clip}\Big(\widehat{d}_h^c, -b_{h,c}^{\text{rem}}, b_{h,c}^{\text{rem}}\Big). \end{aligned} \tag{34}$$

For learned calibration, $\widehat{d}_h^c$ denotes the corresponding network entry $\widehat{d}_{t,h}^{c,\text{req}}$ with t suppressed; teacher requests are specified below. Thus, the chunk budget is enforced separately for radial, tangential, vertical, and yaw corrections, not as a shared four-channel norm constraint. In the nominal configuration, each translational channel has a 1 m step bound and 10 m chunk budget; yaw has a 0.3 rad step bound and 3 rad chunk budget.

Causal geometric teacher. The teacher processes the decoded chunk in order, using the same predicted-state transition and independent correction budgets. Its nominal angular increment is derived from the target radius and the fixed nominal tangential step $\ell_{\tan}^\star = 1\,\text{m/step}$, rather than fixed across radii:

$$\begin{aligned} \alpha^\star(r^\star) &= \arcsin\left(\min\left\{\frac{\ell_{\tan}^\star}{r^\star}, 0.999999\right\}\right), \\ v_\tau^\star &= r^\star \sin\alpha^\star(r^\star). \end{aligned} \tag{35}$$

For the RQ4 radii $r^\star \in \{13, 25, 37\}$ m, the nominal angular increments are approximately 0.076999, 0.040011, and 0.027030 rad, respectively. All three yield $v_\tau^\star = 1\,\text{m/step}$. The same radius-dependent rule is used in teacher-label construction and online analytic evaluation.

This teacher parameter is distinct from the expert trajectory's actual angular increment. The trajectory generator divides one revolution uniformly into N steps, with angular-step magnitude $\Delta\theta^E$:

$$N = \left\lceil \frac{2\pi r^\star}{\ell_{\tan}^\star} \right\rceil, \qquad \Delta\theta^E = \frac{2\pi}{N}. \tag{36}$$

Neither A nor C receives a separate $\alpha^\star$ input field; their geometric condition remains 13-dimensional. The teacher uses $\alpha^\star(r^\star)$ only to set its nominal tangential displacement. That desired displacement is enabled within $\epsilon_{\text{orb}} = 1.5$ m of the target radius:

$$\widehat{d}_h^\tau = \mathbb{I}\big[\big|\|\mathbf{p}_h\|_2 - r^\star\big| \le \epsilon_{\text{orb}}\big]\, v_\tau^\star - \Pi_{xy}(\mathbf{a}_h^S)^\top \mathbf{u}_h^\tau. \tag{37}$$

Outside this band, the request cancels the base action's tangential component, subject to its budget; the tangential correction itself is not switched off. The bounded tangential correction is computed before the radial request. Set

$$\begin{aligned} \mathbf{q}_h &= \mathbf{p}_h + \Pi_{xy}(\mathbf{a}_h^S) + d_h^\tau \mathbf{u}_h^\tau, \\ a_h &= \mathbf{q}_h^\top \mathbf{u}_h^r, \qquad\qquad b_h^2 = \max\{\|\mathbf{q}_h\|_2^2 - a_h^2, 0\}, \\ D_h &= (r^\star)^2 - b_h^2. \end{aligned} \tag{38}$$

The unbounded radial request minimizes next-step radius error along $\mathbf{u}_h^r$. Define the candidate roots $\mathcal{R}_h = \{-a_h - \sqrt{\max(D_h, 0)}, -a_h + \sqrt{\max(D_h, 0)}\}$. With $\epsilon_D = 10^{-8}\,\text{m}^2$, the request is

$$\widehat{d}_h^r = \begin{cases} -a_h, & D_h < -\epsilon_D, \\ \underset{d \in \mathcal{R}_h}{\arg\min}(|d|, d), & D_h \ge -\epsilon_D. \end{cases} \tag{39}$$

Lexicographic minimization selects the smaller-magnitude root, then the smaller signed root in a tie. If the radial line does not intersect the target circle, $-a_h$ attains the smallest reachable radius. The discriminant tolerance clamps small negative values to zero; the request then passes through the radial correction budget.

The vertical request cancels the predicted next-step height error:

$$\widehat{d}_h^z = z^\star - (z_h + \Delta z_h^S). \tag{40}$$

After bounding the radial and vertical requests, the teacher computes the next horizontal position with Eq. equation 33. It requests yaw correction only when $\big|\|\mathbf{p}_{h+1}\|_2 - r^\star\big| \le \epsilon_{\mathrm{orb}}$. Using the signed tangent at that next position and the implementation's heading convention, the teacher computes the reference heading and yaw request as follows:

$$\begin{aligned} \psi^{\mathrm{ref}}_{h+1} &= \mathrm{wrap}\Big(\frac{\pi}{2} - \mathrm{atan2}(u^{\tau}_{h+1,y}, u^{\tau}_{h+1,x})\Big), \\ \widehat{d}^{\psi}_h &= \mathrm{wrap}\big(\psi^{\mathrm{ref}}_{h+1} - \mathrm{wrap}(\psi_h + \Delta\psi^S_h)\big). \end{aligned} \tag{41}$$

Outside this next-position radius band, $\widehat{d}^{\psi}_h = 0$. The yaw request is also bounded before the predicted state is advanced. Thus, tangential clipping precedes radial geometry, and the bounded horizontal prediction precedes yaw alignment. Every channel passes through Eq. equation 34. The teacher reads current geometry and the sampled chunk, but no future expert states or actions. This is a decoded world-frame kinematic transition, not a physical dynamics model.

Degenerate radius. The radial-axis threshold is $\epsilon_r = 10^{-8}$ m. In the RQ4 online evaluator, a near-zero chunk-onset radius is regularized along the most recent valid radial unit vector $\mathbf{u}^r_{\mathrm{last}}$, rather than rejected:

$$\mathbf{p}_0 \leftarrow \epsilon_{\mathrm{init}} \mathbf{u}^r_{\mathrm{last}}, \qquad \epsilon_{\mathrm{init}} = 10^{-6}\,\mathrm{m}. \tag{42}$$

This initialization guard is distinct from the within-chunk fallback. If a predicted position has $\|\mathbf{p}_h\|_2 \le \epsilon_r$ for $h > 0$, the controller reuses the most recent valid radial unit vector to construct the signed tangent. Once the radius exceeds ϵ_r, the axis is recomputed from $\mathbf{p}_h$. The same convention defines the next-position tangent used for yaw, without introducing an arbitrary new radial axis.

Direction inferred from the sampled action. In action-derived-direction evaluation, the sign is recomputed at every replanning event. Starting from $\mathbf{p}^S_0 = \mathbf{p}_0$ after the initialization guard, propagate the uncorrected horizontal actions and form the canonical counter-clockwise tangent:

$$\begin{aligned} \mathbf{p}^S_{h+1} &= \mathbf{p}^S_h + \Pi_{xy}(\mathbf{a}^S_h), \\ \mathbf{v}_h &= (-u^{r,S}_{h,y}, u^{r,S}_{h,x})^\top. \end{aligned} \tag{43}$$

The radial vector $\mathbf{u}^{r,S}_h$ follows this base rollout with the same last-valid-axis rule, rather than using the corrected trajectory's radial axis. The score and direction are

$$\begin{aligned} \gamma_t &= \sum_{h=0}^{H-1} \Pi_{xy}(\mathbf{a}^S_h)^\top \mathbf{v}_h, \\ \widehat{s}_t &= \begin{cases} +1, & \gamma_t \ge 0, \\ -1, & \gamma_t < 0. \end{cases} \end{aligned} \tag{44}$$

The decision threshold is exactly zero. A zero score is assigned $+1$; nonzero scores retain their sign, including arbitrarily small negative scores. There is no near-zero band that overrides a negative score to $+1$. The estimate depends on both the sampled actions and the initial reference-relative position. It is used in both the calibrator condition and tangential basis and remains fixed within the chunk.

In the mismatched-action intervention, the sign is inferred from the original base chunk being corrected, while the calibrator's action representation comes from the separately sampled peer chunk. The direction condition and correction basis are therefore held fixed between the matched and mismatched calls. The condition-only control retains the direction bit but does not receive the full action sequence.

E.2 Calibrator network

The calibrator retains all four decoded action coordinates at every step; the geometric projections above do not truncate its action input. The condition in Eq. 6 has 13 coordinates, and the complete input is

$$\begin{aligned} \mathbf{z}^S_t &= \mathrm{vec}(\mathbf{A}^S_t) \in \mathbb{R}^{4H}, \\ \mathbf{x}^A_t &= [\boldsymbol{\kappa}_t(s_t); \mathbf{z}^S_t] \in \mathbb{R}^{13+4H}. \end{aligned} \tag{45}$$

At $H = 24$, the action representation has 96 coordinates and the concatenated input has 109. The direction sign is kept discrete and is not standardized. For continuous coordinate j, calibrator statistics are computed from the training split:

$$\mathrm{Std}_{\mathrm{cal}}(\mathbf{x})_j = \frac{x_j - \mu_j^{\mathrm{cal}}}{\sigma_j^{\mathrm{cal}}}. \tag{46}$$

The shared MLP is

$$\begin{aligned} \mathbf{h}_1 &= \mathrm{SiLU}\left(\mathbf{W}_1 \,\mathrm{Std}_{\mathrm{cal}}(\mathbf{x}_t^A) + \mathbf{b}_1\right), \\ \mathbf{h}_2 &= \mathrm{SiLU}\left(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2\right), \\ \hat{\mathbf{d}}_t^{\mathrm{req}} &= \mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3. \end{aligned} \tag{47}$$

Both hidden layers have width 128. The linear head emits $4H = 96$ correction requests, reshaped into an $H \times 4$ array for the radial, tangential, vertical, and yaw channels. Budget clipping is applied after prediction, not by the linear output head itself.

E.3 Training objectives and information access

Base-policy objective. For normalized expert actions $\widetilde{\mathbf{A}}_t^E$, Gaussian noise $\boldsymbol{\epsilon}$, and flow time τ, the flow-matching path is

$$\mathbf{x}_\tau = \tau\boldsymbol{\epsilon} + (1-\tau)\widetilde{\mathbf{A}}_t^E, \qquad \mathbf{v}^\star = \boldsymbol{\epsilon} - \widetilde{\mathbf{A}}_t^E. \tag{48}$$

The base policy uses its original objective (Black et al., 2024):

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}\big[\big\|\mathbf{v}_\theta(\mathbf{x}_\tau, \tau; O^{\mathrm{ref}}, O_t, \bar{\mathbf{S}}_t, L^{\mathrm{mask}}) - \mathbf{v}^\star\big\|_F^2\big]. \tag{49}$$

All parameters of the base policy, including the VLM and action expert within θ, are optimized during policy fine-tuning. The entire fine-tuned base policy is frozen before calibrator fitting.

Direction source during training and inference. Write $\boldsymbol{\kappa}_t(s)$ for the geometric condition with last entry s. During training, the demonstrated direction s_t^T is used both in $\boldsymbol{\kappa}_t(s_t^T)$ and in the signed basis for teacher supervision:

$$\mathbf{D}_t^T = \mathcal{T}\left(\mathbf{A}_t^S; \mathbf{P}_r, \mathbf{S}_t, r^\star, h^\star, s_t^T\right). \tag{50}$$

The teacher uses the nominal world-frame transition and the same correction budgets as the learned pathway. It does not use future expert actions. The calibrator is fitted to these bounded targets with Eq. equation 8.

At online inference the learned parameters and normalization statistics remain fixed. Only the source of the direction input changes to the estimate from the sampled chunk. The learned requests and analytic corrections are

$$\begin{aligned} \widehat{\mathbf{D}}_t^{\mathrm{req}} &= \mathrm{reshape}_{H\times 4}\, f_\phi\big(\mathrm{Std}_{\mathrm{cal}}[\boldsymbol{\kappa}_t(\widehat{s}_t); \mathrm{vec}(\mathbf{A}_t^S)]\big), \\ \mathbf{D}_t^{\mathrm{analytic}} &= \mathcal{T}\left(\mathbf{A}_t^S; \mathbf{P}_r, \mathbf{S}_t, r^\star, h^\star, s_t^T\right). \end{aligned} \tag{51}$$

The learned requests undergo the clipping in Eq. equation 34. The analytic reference instead executes the bounded teacher rule directly and retains the generator-provided direction. Nominal kinematics and correction budgets are shared, not information available exclusively to ANALYTIC. The two pathways differ in direction access and in learned prediction versus direct analytic computation; they are not information-matched baselines.

E.4 Calibration controls

For context–target group i and sampled chunk k, the complete ordered action representation is

$$\begin{aligned} \mathbf{A}_{i,k}^S &= [\mathbf{a}_{i,k,0}^S, \ldots, \mathbf{a}_{i,k,H-1}^S], \\ \mathbf{z}_{i,k}^S &= \mathrm{vec}(\mathbf{A}_{i,k}^S) \in \mathbb{R}^{4H}. \end{aligned} \tag{52}$$

Each decoded action retains $[\Delta x, \Delta y, \Delta z, \Delta\psi]^\top$; no translational projection is applied before vectorization. At $H = 24$, $\mathbf{z}^S_{i,k}$ has 96 coordinates. The condition-only control C receives the 13-dimensional geometric condition, whereas A receives the $13 + 4H = 109$-dimensional concatenation. The two models predict

$$\widetilde{\mathbf{d}}^C_i = f_C(\boldsymbol{\kappa}^{\mathrm{cal}}_i), \qquad \widetilde{\mathbf{d}}^A_{i,k} = f_A(\boldsymbol{\kappa}^{\mathrm{cal}}_i, \mathbf{z}^S_{i,k}). \tag{53}$$

For B chunks sampled under the same condition, the scalar teacher-correction variation is

$$V_i = \sqrt{\frac{1}{H}\sum_{h=0}^{H-1} \mathrm{Var}_{k=1,\dots,B}[d^T_{i,k,h}]}. \tag{54}$$

Here B counts sampled chunks and K remains the execution-prefix length. Finally, the matched–mismatched intervention applies

$$\begin{aligned} \widetilde{\mathbf{d}}^{\mathrm{match}}_{i,k} &= f_A(\boldsymbol{\kappa}^{\mathrm{cal}}_i, \mathbf{z}^S_{i,k}), \\ \widetilde{\mathbf{d}}^{\mathrm{mismatch}}_{i,k} &= f_A(\boldsymbol{\kappa}^{\mathrm{cal}}_i, \mathbf{z}^S_{i,\pi_i(k)}), \end{aligned} \tag{55}$$

where $\pi_i(k) \neq k$. Both corrections are applied to the same base chunk; only the $4H$-dimensional action representation read by A changes.

F Cross-VLA Command Intervention

We evaluate OpenVLA, TinyVLA, Octo-UAV, π_0, and SmolVLA under controlled numerical command interventions. For each model, we hold the visual observation, native UAV state, semantic instruction context, and inference configuration fixed while varying only the commanded physical quantity.

Because action interfaces differ across architectures, comparisons are aligned by physical execution horizon rather than action-token count. The same reference-relative metric definitions and kinematic rollout evaluator are used for all models.

G Full-Trajectory Closed-Loop Evaluation

We evaluate complete recovery-and-orbit trajectories through closed-loop visual re-observation under an additive world-frame kinematic transition model. The evaluator integrates decoded displacement actions but does not model inertia, aerodynamics, actuator dynamics, contact, or collision validity; the results therefore do not constitute physics-simulator or real-flight validation.

At each replanning state, the policy receives the episode reference image, the current rendered observation, the current UAV state, and the recomputed numerical interface. It samples an $H = 24$ action chunk, applies calibration when applicable, executes a fixed six-action prefix, and then re-renders the current observation before replanning. The reference remains fixed throughout the rollout, while $y_{q,t}$ and $e_{q,t}$ are recomputed after each executed prefix.

For both tasks, initial task-metric offsets are drawn from

$$\{-4, -2, 0, +2, +4\}\ \mathrm{m}. \tag{56}$$

Orbit-R rollouts recover toward the commanded radius and then maintain that radius through one signed orbit around the reference. Orbit-H rollouts recover toward the commanded relative height and maintain that height through the corresponding orbit. Contexts, offsets, sampling seeds, and qualitative examples are selected before confirmation outcomes are inspected.

We compare the base policy, C, A-Mismatch, A, Analytic-ActionDir, and the analytic oracle in Table 2. Analytic-ActionDir directly applies the geometric rule with the same ActionDir estimator, geometry inputs, execution protocol, and channel budgets as A; it does not receive generator-provided direction. This matches the estimation rule and available information, not necessarily the realized direction estimates after their closed-loop trajectories diverge. Analytic uses generator-provided direction and remains a privileged reference. All comparisons use the decoded-action kinematic transition; none constitutes physical-flight validation.

Qualitative closed-loop rollouts. Figure 5 illustrates the recovery-and-orbit trajectories, angular progress, and target errors accompanying the main-text quantitative comparison.

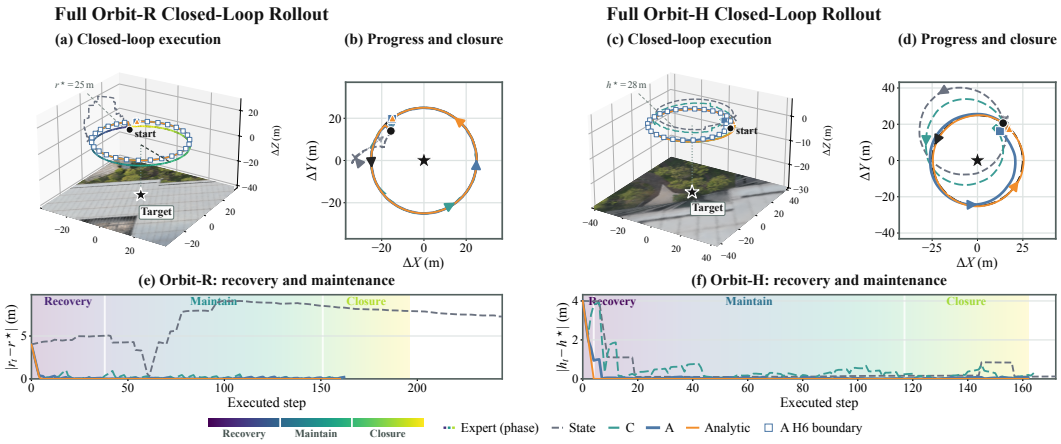


Figure 5: **Closed-loop recovery and orbit execution.** Panels (a,c) show reference-centered 3D trajectories, (b,d) angular progress and closure, and (e,f) radius and height errors around recovery under 3DGS visual feedback, with square markers indicating A's K-action replanning boundaries.

G.1 Metric Computation and Temporal Reference Points

Expert-referenced XYZ RMSE for action chunks. For an evaluated H-step chunk i, calibrated and expert world-frame positions are reconstructed from the same initial state by cumulatively summing their respective translational actions. Let $\mathbf{x}_{i,h}^{W}$ and $\mathbf{x}_{i,h}^{E,W}$ denote the resulting 3D positions after h actions, for $h = 1, \ldots, H$. The coordinate-wise RMSE is first computed separately for each chunk:

$$e_i^{\mathrm{xyzRMSE}} = \sqrt{\frac{1}{3H} \sum_{h=1}^{H} \left\| \mathbf{x}_{i,h}^{W} - \mathbf{x}_{i,h}^{E,W} \right\|_2^2}. \tag{57}$$

The divisor counts scalar position coordinates, $3H$, and excludes the initial position at $h = 0$. There is no normalization by target radius, trajectory length, or scene scale; the result is in meters. This differs from the mean per-step Euclidean distance $E_{\mathrm{traj3D},q}$ in Eq. equation 17.

The confirmation result averages the already-computed chunk errors within each held-out landmark, then weights the four landmarks equally. Let $\mathcal{E}_\ell$ contain the evaluated chunks for landmark ℓ, including targets, phases, action samples, and training seeds. At $H = 24$, the result is

$$E_{\mathrm{xyzRMSE}}^{H24} = \frac{1}{4} \sum_{\ell=1}^{4} \frac{1}{|\mathcal{E}_\ell|} \sum_{i \in \mathcal{E}_\ell} e_i^{\mathrm{xyzRMSE}}. \tag{58}$$

This is a mean of per-chunk RMSE values, not one square root taken after pooling squared errors across chunks. Confidence intervals are computed by resampling the four landmark means; training seeds are included within each landmark's chunk set, not introduced as an additional outer equal-weight average.

Joint entry and post-recovery metrics. We next define quantities for one closed-loop rollout and suppress its index i. Let $\mathbf{S}_0, \ldots, \mathbf{S}_T$ be its states, with T the final executed step. Radius and relative height are $r_t = \|\mathbf{S}_{t,xy} - \mathbf{P}_{r,xy}\|_2$ and $h_t = S_{t,z} - P_{r,z}$. The first simultaneous radius–height entry time is

$$\tau_{\mathrm{rec}} = \min \left\{ t \in \{0, \ldots, T\} : \begin{array}{l} |r_t - r^\star| \le 1\,\mathrm{m}, \\ |h_t - h^\star| \le 1\,\mathrm{m} \end{array} \right\}. \tag{59}$$

Both inequalities must hold at the same state. Thus, joint entry is not obtained by intersecting two indicators of having recovered at possibly different times. If the set is empty, write $\tau_{\mathrm{rec}} = \infty$ and joint recovery fails. The initial state can qualify, giving $\tau_{\mathrm{rec}} = 0$.

When $\tau_{\mathrm{rec}} < T$, maintenance errors exclude the entry state and average only the subsequent states:

$$E_{\mathrm{R}}^{\mathrm{post}} = \frac{1}{T - \tau_{\mathrm{rec}}} \sum_{t=\tau_{\mathrm{rec}}+1}^{T} |r_t - r^\star|,$$
$$E_{\mathrm{H}}^{\mathrm{post}} = \frac{1}{T - \tau_{\mathrm{rec}}} \sum_{t=\tau_{\mathrm{rec}}+1}^{T} |h_t - h^\star|. \tag{60}$$

Let φ_t be the unwrapped polar angle around the reference center. Post-recovery angular progress and three-dimensional closure are

$$\Theta^{\mathrm{post}} = \varphi_T - \varphi_{\tau_{\mathrm{rec}}},$$
$$D_{\mathrm{close}}^{\mathrm{post}} = \|\mathbf{S}_{T,xyz} - \mathbf{S}_{\tau_{\mathrm{rec}},xyz}\|_2. \tag{61}$$

The post-recovery angular-error diagnostic is $E_\theta^{\mathrm{post}} = \big||\Theta^{\mathrm{post}}| - 2\pi\big|$. It does not by itself test the required orbit direction. With $s \in \{-1, +1\}$ denoting that evaluation direction, full-lap success requires all four conditions

$$|\Theta^{\mathrm{post}} - s\,2\pi| \le \pi/12,$$
$$E_{\mathrm{R}}^{\mathrm{post}} \le 1\,\mathrm{m}, \qquad E_{\mathrm{H}}^{\mathrm{post}} \le 1\,\mathrm{m}, \tag{62}$$
$$D_{\mathrm{close}}^{\mathrm{post}} \le 5\,\mathrm{m}.$$

The angular tolerance is $\pi/12$ rad, or 15°. The required direction s is used for evaluation, not substituted for the learned calibrator's action-derived $\widehat{s}_t$.

If joint entry never occurs, or occurs only at the terminal state, post-recovery metrics are undefined. These rollouts are excluded from post-recovery means and counted as unsuccessful for full-lap success. Joint-entry coverage is reported separately, including terminal-only entry. To state the aggregation, restore rollout indices and let $\mathcal{V} = \{i : \tau_{\mathrm{rec},i} < T_i\}$ be the valid post-entry set. For $q \in \{\mathrm{R}, \mathrm{H}\}$, the maintenance values in joint tables are

$$E_{\mathrm{maint},q} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} E_{i,q}^{\mathrm{post}}. \tag{63}$$

An empty $\mathcal{V}$ yields an undefined mean, not zero. With $L_i = 1$ only when $i \in \mathcal{V}$ and all conditions in Eq. equation 62 hold, joint coverage and full-lap success are

$$M_{\mathrm{rec}}^{\mathrm{joint}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[\tau_{\mathrm{rec},i} \le T_i],$$
$$M_{\mathrm{lap}}^{\mathrm{joint}} = \frac{1}{N} \sum_{i=1}^{N} L_i. \tag{64}$$

Both denominators include every evaluated rollout. Entries shown as percentages display 100 times these rates.

Full-trajectory geometry. The separate full-trajectory diagnostic starts at the rollout's offset initial state $\mathbf{S}_0$ and includes the recovery phase. For one rollout, direction-aligned progress and angular error are

$$\Theta^{\mathrm{full}} = s(\varphi_T - \varphi_0),$$
$$E_\theta^{\mathrm{full}} = |\Theta^{\mathrm{full}} - 2\pi|. \tag{65}$$

Its closure reference is the source expert's lap-start position, $\mathbf{S}^E_{t_{\mathrm{lap}},xyz}$:

$$D_{\mathrm{close}}^{\mathrm{full}} = \|\mathbf{S}_{T,xyz} - \mathbf{S}^E_{t_{\mathrm{lap}},xyz}\|_2. \tag{66}$$

That reference generally differs from both $\mathbf{S}_0$ and $\mathbf{S}_{\tau_{\mathrm{rec}}}$. Full-trajectory angular completion checks whether the direction-aligned progress reaches the one-lap tolerance at any executed state:

$$M_{\mathrm{comp}}^{\mathrm{full}} = \mathbf{1}\left[\max_{0 \le t \le T} s(\varphi_t - \varphi_0) \ge 2\pi - \pi/12\right]. \tag{67}$$

It does not impose radius, height, or spatial-closure constraints, and is not the joint full-lap success indicator L_i. These full-trajectory diagnostics are defined for recovered and non-recovered rollouts alike.

Restoring rollout indices, we compute the aggregate full-trajectory geometry metrics as follows:

$$E_\Theta = \frac{1}{N}\sum_{i=1}^{N} E_{\theta,i}^{\text{full}},$$
$$D_{\text{close}} = \frac{1}{N}\sum_{i=1}^{N} D_{\text{close},i}^{\text{full}}. \tag{68}$$

The 5 m success threshold applies to $D_{\text{close}}^{\text{post}}$, not to $D_{\text{close}}^{\text{full}}$. Post-entry maintenance, post-entry closure, and full-trajectory geometry therefore must retain their distinct time and position references. Seed-wise summaries use the equal-seed aggregation specified in Appendix D. The H24 confirmation exception is defined in Eq. equation 58; the equal-context estimators for correction-channel and numerical-target evaluations are specified in Appendices G.4 and G.2, respectively.

We additionally report the official HUGE-Bench metrics TCR, nDTW, NTP, SR, CR, and CSPL on `test_seen` and `test_unseen`. The collision-dependent CR and CSPL are omitted when aligned collision information is unavailable; they are not redefined here.

G.2 Same-State Numerical-Target Generalization

Targets and paired evaluation. The policy-adaptation and calibrator-training target supports are those in Table 3. For the additional closed-loop evaluation in RQ4, we vary $r^\star \in \{25, 20, 10, 43\}$ m at $h^\star = 40$ m, and $h^\star \in \{40, 34, 24, 60\}$ m at $r^\star = 25$ m. In each list, the values correspond to seen, unseen in-range, low-extrapolation, and high-extrapolation conditions. The seen pair $(25, 40)$ is shared: the eight axis–target presentations comprise seven distinct target pairs, not eight independent sets of trajectories. This protocol is separate from the collection-grid interpolation evaluation in Table 3.

The four held-out physical landmarks have identifiers 6, 8, 70, and 72. Each has five paired initial states, with nominal radius–height offsets relative to the seen pair of $(-4, +4)$, $(-2, -2)$, $(0, 0)$, $(+2, +2)$, and $(+4, -4)$ m. Within each landmark–initial-state group, target interventions share the initial UAV state, initial RGB observation, canonical landmark-reference RGB image, and masked semantic instruction template. Action-sampling noise is paired by training seed, context, sample identifier, and replanning index. Only the requested scalar and its derived numerical state condition change at initialization. After trajectories diverge, observations, states, and base chunks are recomputed separately for each method and target; paired noise does not make these chunks identical.

All checkpoints, correction bounds, and calibration settings remain frozen. Three independently trained systems (seeds 33–35) yield 60 rollouts per method and target pair. The 20 initial contexts crossed with seven target pairs give 140 state–target combinations, 420 rollouts per method, and 2,100 rollouts across five methods. We use $H = 24$, $K = 6$, 3DGS-rendered visual re-observation, and additive world-frame kinematic updates. The budget rule is shared: a rollout executes at most the smaller of 512 actions and the corresponding expert-lap length plus 48 actions. Thus, the rule, not necessarily the numerical cap, is fixed across targets. Success or a radial safety violation terminates a rollout earlier.

Method and measurement contracts. We retain the main-text shorthand: C, A-MISMATCH, A, and ANALYTIC correspond to C-ACTIONDIR, A-M-ACTIONDIR, A-ACTIONDIR, and FULL-ANALYTIC, respectively. STATE executes the sampled base chunk without calibration. C conditions on the numerical context and the direction bit inferred from the current base chunk, but not its ordered action sequence. A uses the matched chunk. A-MISMATCH predicts correction requests from an independently sampled same-state peer chunk and applies them to the original chunk; its direction bit is still inferred from the original chunk being corrected. Direction is re-estimated at each replanning step. None of the learned conditions uses expert direction at inference. ANALYTIC does, so it is an oracle geometric reference rather than an equal-information baseline.

The $M_{\mathrm{rec}}^{\mathrm{joint}}$ and $M_{\mathrm{lap}}^{\mathrm{joint}}$ criteria follow Appendix G.1: joint 1-m target bands, post-entry errors at most 1 m, signed angular progress within 15° of the commanded lap, and $D_{\mathrm{close}}^{\mathrm{post}} \leq 5$ m. We report post-entry errors as $E_{\mathrm{maint},q}$ and terminal errors as $E_{\mathrm{final},q}$. The additional geometric diagnostics are $E_{\theta}^{\mathrm{post}} = \left||\Theta^{\mathrm{post}}| - 2\pi\right|$ and $D_{\mathrm{close}}^{\mathrm{post}}$, referenced to joint entry, not the all-rollout E_{Θ} or source-expert-referenced D_{close}. The angular diagnostic is unsigned; the success test retains the commanded sign. Post-entry quantities are undefined without a valid post-entry segment, whereas $E_{\mathrm{final},q}$ uses all rollouts.

Aggregation and uncertainty. Tables 5–6 and 7–8 report recovery, maintenance, and geometric results for all eight targets. For each target pair, we average over the five states and three systems within each landmark, then weight the four landmarks equally. Conditional metrics use eligible rollouts within each landmark; undefined values are not zero errors. This population and estimator are distinct from the 180-rollout, equal-seed main comparison in Table 2; the seen rows do not replace that comparison. The paired intervals in Table 9 enumerate all $4^4 = 256$ ordered resamples of four landmarks with replacement, using the same resampled landmarks for both conditions. With only four landmark clusters, these 95% intervals are descriptive, not evidence of broad scene coverage. Fig. 4 shows means without method-wise intervals; paired-difference intervals are not substituted for such intervals. In Fig. 4, shading marks the training-target range rather than continuous training coverage, and connecting lines only guide the eye.

Table 5: **Generalization across radius targets at $h^\star = 40$ m: recovery and maintenance.**

Target (m)	Method	$M_{\mathrm{rec}}^{\mathrm{joint}}$ (%) ↑	$M_{\mathrm{lap}}^{\mathrm{joint}}$ (%) ↑	$E_{\mathrm{maint,R}}$ (m) ↓	$E_{\mathrm{maint,H}}$ (m) ↓
25	STATE	35.0	1.7	2.692	0.217
	C	100.0	80.0	0.358	0.294
	A-MISMATCH	100.0	100.0	0.116	0.025
	A	100.0	100.0	0.035	0.021
	ANALYTIC	100.0	100.0	< 0.001	< 0.001
20	STATE	18.3	3.3	2.368	0.374
	C	100.0	86.7	0.425	0.299
	A-MISMATCH	100.0	100.0	0.127	0.030
	A	100.0	100.0	0.041	0.021
	ANALYTIC	100.0	100.0	< 0.001	< 0.001
10	STATE	0.0	0.0	–	–
	C	96.7	46.7	0.968	0.200
	A-MISMATCH	100.0	93.3	0.440	0.072
	A	100.0	93.3	0.274	0.069
	ANALYTIC	100.0	100.0	< 0.001	< 0.001
43	STATE	0.0	0.0	–	–
	C	100.0	76.7	0.474	0.394
	A-MISMATCH	100.0	80.0	0.161	0.031
	A	100.0	81.7	0.066	0.025
	ANALYTIC	100.0	100.0	< 0.001	< 0.001

Notes. Recovery and lap columns report percentages; maintenance errors are in meters. A dash denotes no qualifying joint-entry trajectory.

Paired effects and interpretation. Differences in Table 9 use matched seed–initial-state–target pairs. Negative $\Delta E_{\mathrm{maint},q}$ and positive $\Delta M_{\mathrm{lap}}^{\mathrm{joint}}$ favor A. The former uses pairs with valid post-entry segments for both methods. At $r^\star = 10$ m, the A–C comparison has 58 rather than 60 such pairs because two C rollouts never enter joint tolerance. Its paired effect therefore need not equal the difference between the separately aggregated, rounded means.

Across all eight reported axis–target conditions, A has lower paired $E_{\mathrm{maint},q}$ on the varied axis than C and A-MISMATCH. In contrast, A and A-MISMATCH tie on $M_{\mathrm{lap}}^{\mathrm{joint}}$ at the seen and in-range targets; their extrapolation differences are small relative to the four-landmark intervals and do not establish a general completion advantage or equivalence. For A, $r^\star = 43$ m gives $D_{\mathrm{close}}^{\mathrm{post}} = 15.402$ m despite $E_{\mathrm{maint,R}} = 0.066$ m. At $h^\star = 24$ and 60 m, $M_{\mathrm{lap}}^{\mathrm{joint}}$ is 93.3% and 100%, respectively, with $E_{\mathrm{maint,H}} = 0.029$ and 0.027 m. These outcomes distinguish numerical maintenance from

Table 6: **Generalization across radius targets at $h^\star = 40$ m: terminal and post-entry geometry.**

Target (m)	Method	$E_{\text{final,R}}$ (m) ↓	$E_{\text{final,H}}$ (m) ↓	E_θ^{post} (rad) ↓	$D_{\text{close}}^{\text{post}}$ (m) ↓
	STATE	3.986	1.894	4.674	19.660
	C	0.293	0.323	0.618	9.758
25	A-MISMATCH	0.124	0.021	0.179	4.463
	A	0.032	0.019	0.183	4.566
	ANALYTIC	< 0.001	< 0.001	0.163	4.090
	STATE	5.605	1.709	3.901	17.263
	C	0.393	0.316	0.492	6.023
20	A-MISMATCH	0.096	0.022	0.221	4.525
	A	0.026	0.023	0.222	4.536
	ANALYTIC	< 0.001	< 0.001	0.232	4.714
	STATE	15.172	2.411	–	–
	C	1.361	0.215	1.911	8.008
10	A-MISMATCH	0.376	0.052	0.477	3.033
	A	0.174	0.044	0.501	2.383
	ANALYTIC	< 0.001	< 0.001	0.175	1.912
	STATE	17.504	2.402	–	–
	C	0.564	0.402	1.034	13.484
43	A-MISMATCH	0.198	0.032	0.526	15.952
	A	0.047	0.017	0.677	15.402
	ANALYTIC	< 0.001	< 0.001	0.099	4.262

Notes. Terminal and closure errors are in meters; the post-entry angle is in radians. A dash denotes no qualifying joint-entry trajectory. ANALYTIC is the expert-direction oracle.

Table 7: **Generalization across height targets at $r^\star = 25$ m: recovery and maintenance.**

Target (m)	Method	$M_{\text{rec}}^{\text{joint}}$ (%) ↑	$M_{\text{lap}}^{\text{joint}}$ (%) ↑	$E_{\text{maint,R}}$ (m) ↓	$E_{\text{maint,H}}$ (m) ↓
	STATE	35.0	1.7	2.692	0.217
	C	100.0	80.0	0.358	0.294
40	A-MISMATCH	100.0	100.0	0.116	0.025
	A	100.0	100.0	0.035	0.021
	ANALYTIC	100.0	100.0	< 0.001	< 0.001
	STATE	31.7	1.7	3.099	0.424
	C	100.0	85.0	0.374	0.336
34	A-MISMATCH	100.0	100.0	0.122	0.042
	A	100.0	100.0	0.039	0.023
	ANALYTIC	100.0	100.0	< 0.001	< 0.001
	STATE	23.3	0.0	3.374	0.426
	C	100.0	68.3	0.526	0.276
24	A-MISMATCH	100.0	91.7	0.148	0.048
	A	100.0	93.3	0.055	0.029
	ANALYTIC	100.0	100.0	< 0.001	< 0.001
	STATE	15.0	0.0	2.743	0.574
	C	100.0	83.3	0.447	0.290
60	A-MISMATCH	100.0	100.0	0.144	0.051
	A	100.0	100.0	0.049	0.027
	ANALYTIC	100.0	100.0	< 0.001	< 0.001

Notes. Recovery and lap columns report percentages; maintenance errors are in meters. A dash denotes no qualifying joint-entry trajectory.

Table 8: **Generalization across height targets at $r^\star = 25$ m: terminal and post-entry geometry.**

Target (m)	Method	$E_{\mathrm{final,R}}$ (m) ↓	$E_{\mathrm{final,H}}$ (m) ↓	E_θ^{post} (rad) ↓	$D_{\mathrm{close}}^{\mathrm{post}}$ (m) ↓
40	STATE	3.986	1.894	4.674	19.660
	C	0.293	0.323	0.618	9.758
	A-MISMATCH	0.124	0.021	0.179	4.463
	A	0.032	0.019	0.183	4.566
	ANALYTIC	< 0.001	< 0.001	0.163	4.090
34	STATE	3.961	2.539	5.094	19.163
	C	0.331	0.402	0.727	8.590
	A-MISMATCH	0.127	0.022	0.178	4.491
	A	0.036	0.020	0.181	4.552
	ANALYTIC	< 0.001	< 0.001	0.163	4.104
24	STATE	3.876	5.714	6.016	8.339
	C	0.449	0.266	1.443	12.089
	A-MISMATCH	0.098	0.021	0.251	6.102
	A	0.035	0.018	0.270	6.490
	ANALYTIC	< 0.001	< 0.001	0.163	4.094
60	STATE	3.924	9.415	6.142	5.541
	C	0.229	0.313	0.748	6.916
	A-MISMATCH	0.091	0.020	0.179	4.511
	A	0.031	0.023	0.178	4.496
	ANALYTIC	< 0.001	< 0.001	0.163	4.101

Notes. Terminal and closure errors are in meters; the post-entry angle is in radians. A dash denotes no qualifying joint-entry trajectory. ANALYTIC is the expert-direction oracle.

Table 9: **Paired numerical-target comparisons for A.** Paired maintenance-error differences favor A over both comparators, whereas the completion intervals do not establish an advantage over A-MISMATCH.

$\Delta M_{\mathrm{lap}}^{\mathrm{joint}}$ **(percentage points)** ↑

Axis	Target (m)	C	A-MISMATCH
Radius	25	$+20.0\,[+3.3, +36.7]$	$0.0\,[0.0, 0.0]$
Radius	20	$+13.3\,[+3.3, +26.7]$	$0.0\,[0.0, 0.0]$
Radius	10	$+46.7\,[+41.7, +51.7]$	$0.0\,[-5.0, +5.0]$
Radius	43	$+5.0\,[+1.7, +6.7]$	$+1.7\,[-3.3, +6.7]$
Height	40	$+20.0\,[+3.3, +36.7]$	$0.0\,[0.0, 0.0]$
Height	34	$+15.0\,[0.0, +30.0]$	$0.0\,[0.0, 0.0]$
Height	24	$+25.0\,[+3.3, +46.7]$	$+1.7\,[0.0, +5.0]$
Height	60	$+16.7\,[+6.7, +26.7]$	$0.0\,[0.0, 0.0]$

$\Delta E_{\mathrm{maint},q}$ **(m)** ↓

Axis	Target (m)	C	A-MISMATCH
Radius	25	$-0.323\,[-0.377, -0.281]$	$-0.081\,[-0.090, -0.065]$
Radius	20	$-0.384\,[-0.444, -0.323]$	$-0.086\,[-0.099, -0.076]$
Radius	10	$-0.692\,[-0.756, -0.618]$	$-0.166\,[-0.200, -0.125]$
Radius	43	$-0.408\,[-0.467, -0.350]$	$-0.095\,[-0.106, -0.085]$
Height	40	$-0.274\,[-0.303, -0.233]$	$-0.0046\,[-0.0065, -0.0025]$
Height	34	$-0.312\,[-0.327, -0.298]$	$-0.0182\,[-0.0279, -0.0101]$
Height	24	$-0.247\,[-0.262, -0.232]$	$-0.0191\,[-0.0276, -0.0101]$
Height	60	$-0.263\,[-0.279, -0.243]$	$-0.0232\,[-0.0323, -0.0142]$

Notes. Each cell reports the equal-landmark mean and descriptive 95% exact four-landmark cluster-bootstrap interval for A minus the comparator. $\Delta M_{\mathrm{lap}}^{\mathrm{joint}}$ is in percentage points; $\Delta E_{\mathrm{maint},q}$ is in meters and uses the varied axis. The two seen presentations reuse the same target-pair rollouts.

full-trajectory completion. They establish neither simultaneous radius–height extrapolation nor robustness to non-kinematic execution.

G.3 Action-representation ablation

This subsection reports offline replay and full-trajectory comparisons of action representations. They remain separate from the nominal RQ4 comparison and the frozen-checkpoint channel-mask evaluation in Appendix G.4. Their metrics and evaluation populations are identified separately below.

Offline action-representation comparison. To test whether compact action summaries can replace the complete sequence, we compare the first action, the channel-wise chunk sum, a shuffled sequence, and the full ordered chunk in single-chunk replay. This comparison is distinct from the 24-step feedback experiment in RQ3.

Table 10: **Offline action-representation ablation.** Full ordered inputs achieve the lowest observed radius and height MAEs, with XYZ RMSE close to that of shuffled inputs.

Representation	$E_{\mathrm{metric,R}}$ (m) ↓	$E_{\mathrm{metric,H}}$ (m) ↓	$E^{H24}_{\mathrm{xyzRMSE}}$ (m) ↓
C	1.993	0.170	3.601
$\mathrm{A}_{\mathrm{first}}$	1.285	0.110	3.227
$\mathrm{A}_{\mathrm{end}}$	1.249	0.119	3.203
$\mathrm{A}_{\mathrm{shuf}}$	1.218	0.113	3.189
A	**1.192**	**0.078**	**3.188**

The first-action and chunk-sum representations are denoted $\mathrm{A}_{\mathrm{first}}$ and $\mathrm{A}_{\mathrm{end}}$; $\mathrm{A}_{\mathrm{shuf}}$ retains the actions but changes their temporal order. C receives only the calibration condition, whereas A receives the complete ordered chunk. Table 10 reports trajectory radius and height MAEs and expert-referenced coordinate-wise XYZ RMSE. These are offline action-chunk metrics, not closed-loop terminal errors or conditional maintenance MAEs. First-action and endpoint summaries capture most of the gain over C. The full ordered chunk has the lowest observed radius and height errors, while its XYZ RMSE is close to that of the shuffled input. The representation comparison is also separate from the five-method H24 confirmation in Appendix I.1; their means should not be substituted for one another.

Closed-loop representation tables. Table 11 reports full-trajectory terminal accuracy, while Table 12 reports post-recovery maintenance and closure. These are separate from the offline action-chunk metrics in Table 10.

Table 11: **Closed-loop action representations: terminal accuracy.** Full-lap success and terminal errors.

Representation	M_{lap} (%) ↑	$E_{\mathrm{final,R}}$ (m) ↓	$E_{\mathrm{final,H}}$ (m) ↓
Condition only (C)	81.11	0.405	0.339
First action	79.44	0.124	0.036
Chunk endpoint	81.11	0.218	0.092
Shuffled chunk	**93.33**	0.070	0.043
Full ordered chunk (A)	92.78	**0.045**	**0.026**

Table 12: **Closed-loop action representations: post-recovery maintenance.** Maintenance MAEs and closure distance.

Representation	M_{lap} (%) ↑	E_{R} (m) ↓	E_{H} (m) ↓	D_{close} (m) ↓
Condition only (C)	81.11	0.466	0.282	6.582
First action	79.44	0.146	0.050	7.363
Chunk endpoint	81.11	0.230	0.089	7.708
Shuffled chunk	**93.33**	0.075	0.048	**5.057**
Full ordered (A)	92.78	**0.054**	**0.028**	6.769

Note. All methods achieve 100% joint recovery. Boldface identifies the best observed mean in each metric; it does not imply statistical separation.

Protocol and evaluation population. We keep the six-scalar base policy, geometric teacher, correction bounds, and action-derived orbit-direction bit fixed and train a separate calibrator for each representation. The alternatives use the first translational action, the channel-wise sum of the $H = 24$ actions, or the complete chunk in a deterministic sample-specific shuffled order. C uses only the condition, whereas A uses the full ordered chunk. Seeds 33–35 and four held-out landmarks yield 180 rollouts per method (900 trajectories across the five methods). Reported means weight the four landmarks equally. All five methods enter the joint tolerance band in 100% of rollouts; full-lap success ranges from 79.44% to 93.33%.

Here M_{lap} denotes joint full-lap success, and E_{R} and E_{H} are post-recovery maintenance MAE. The D_{close} column is the closure-distance statistic for this representation comparison, denoted $D_{\mathrm{close}}^{\mathrm{repr}}$ when distinguishing protocols. Its values are not substituted for the main evaluation's closure field without verifying the temporal-reference mapping.

The offline representation comparison is in Table 10. The maintenance and closure results in Table 12 complement the terminal errors in Table 11; they do not replace the separate nominal comparison in Table 2.

Completion versus precision. Full ordered conditioning raises observed lap success from 81.11% for C to 92.78% and has the lowest radius and height maintenance MAEs (0.054 and 0.028 m). First-action and endpoint summaries have lower lap success and larger maintenance errors than the full chunk. Shuffling retains a similar observed lap-success rate (93.33%) and yields a smaller closure distance (5.057 versus 6.769 m). Thus, improved numerical regulation does not establish that temporal ordering is necessary for completing a lap.

Paired numerical differences. Relative to shuffled conditioning, full ordered conditioning reduces radius and height maintenance MAE by 0.021 and 0.020 m, respectively. The 95% paired landmark-cluster bootstrap intervals for these reductions are [0.015, 0.030] and [0.017, 0.023] m. These intervals support an improvement in numerical regulation; they are not confidence intervals for the lap-success or closure-distance comparisons.

Endpoint and mean pooling. For the additive action representation at fixed horizon, $\mathbf{z}_{\mathrm{end}} = \sum_{h=0}^{H-1} \mathbf{a}_h = H\,\overline{\mathbf{a}}$. At $H = 24$, the channel-wise chunk sum and mean-pooled inputs therefore differ only by a constant scale and are affine-equivalent after corresponding training-split feature standardization. We omit the redundant mean-pooled row from the closed-loop table.

G.4 Correction-channel ablation: protocol and post-entry diagnostics

Frozen-checkpoint intervention. The channel ablation in RQ4 (Section 4.5) uses the RQ4 base policy and calibrator under ActionDir for full-trajectory evaluation. The five masks in Table 13 select channels in $[r, \tau, z, \psi]$ order after training; no checkpoint, correction bound, or learned parameter is changed. A-Full is A-ActionDir, corresponding to A in the main RQ4 comparison. At each replanning step, $\widehat{s}_t$ is computed from the unmodified $H = 24$ base chunk and is shared by the condition and tangential basis. All masks retain the same channel-specific translation and yaw budgets. They use the same 3DGS visual re-observation and additive world-frame transition

protocols. Realized states and observations can differ after execution because the correction masks change the actions; they are not forced to match across diverging rollouts.

Population and aggregation. There are four landmark-disjoint physical contexts, three training seeds (33–35), and 180 rollouts per condition. For a metric m, let $\mathcal{E}_\ell$ contain the evaluated rollouts for context ℓ across targets, initial offsets, and seeds. The reported estimator is

$$\overline{m}_\ell = \frac{1}{|\mathcal{E}_\ell|} \sum_{i \in \mathcal{E}_\ell} m_i, \qquad \overline{m} = \frac{1}{4} \sum_{\ell=1}^{4} \overline{m}_\ell. \tag{69}$$

For conditional post-entry metrics, each context mean uses only qualifying rollouts within that context. Seeds are averaged inside the context, not as an additional outer equal-weight average. The 95% exact cluster-bootstrap intervals enumerate all $4^4 = 256$ ordered resamples of the four context means with replacement. Paired comparisons retain the same resampled contexts for both conditions. This estimator is distinct from the equal-seed aggregation in Table 2; its values should not be substituted into that table. The separate representation ablation in Appendix G.3 is also not this five-mask comparison.

Metric references. For rollout i, the terminal quantities in Table 13 are $|r_{i,T_i} - r_i^\star|$ and $|h_{i,T_i} - h_i^\star|$; their equal-context means are $E_{\mathrm{final,R}}$ and $E_{\mathrm{final,H}}$. These use all rollouts, unlike $E_{\mathrm{maint},q}$. The joint-entry and full-lap criteria are those in Appendix G.1, with the same thresholds. For the post-entry diagnostics, a rollout qualifies only when $\tau_{\mathrm{rec},i} < T_i$; n_{post} counts these rollouts across contexts and seeds. Non-entry and terminal-only entry do not contribute to the conditional means.

$E_{\mathrm{maint},q}$ excludes the entry state. E_θ^{post} summarizes $\big||\Theta^{\mathrm{post}}| - 2\pi\big|$, and $D_{\mathrm{close}}^{\mathrm{post}}$ is referenced to the joint-entry position in three dimensions. $F_{\tan}$ denotes the fraction of post-entry actions with positive tangential progress, aggregated by Eq. equation 69. It is reported as a fraction, not a percentage. The angular diagnostic is unsigned, whereas $M_{\mathrm{lap}}^{\mathrm{joint}}$ tests the required signed direction. These post-entry quantities must not be interpreted as the all-rollout E_Θ and source-expert-referenced D_{close}.

Table 13: **Full-trajectory correction-channel ablation.** Adding yaw correction raises full-lap success, whereas including the tangential channel provides no additional observed completion benefit.

Method	$M_{\mathrm{rec}}^{\mathrm{joint}}$ (%) ↑	$M_{\mathrm{lap}}^{\mathrm{joint}}$ (%) ↑	$E_{\mathrm{final,R}}$ (m) ↓	$E_{\mathrm{final,H}}$ (m) ↓
STATE	38.89 [37.78, 40.00]	2.78 [0.56, 5.56]	3.786 [3.732, 3.851]	1.791 [1.724, 1.831]
A-R	97.22 [96.11, 97.78]	42.78 [35.00, 50.00]	0.130 [0.100, 0.161]	0.329 [0.276, 0.382]
A-RH	100.00 [100.00, 100.00]	52.78 [46.67, 58.89]	0.115 [0.093, 0.137]	0.033 [0.030, 0.037]
A-RH+YAW	100.00 [100.00, 100.00]	93.89 [87.78, 100.00]	0.032 [0.027, 0.039]	0.026 [0.020, 0.034]
A-FULL	100.00 [100.00, 100.00]	93.33 [87.78, 98.89]	0.043 [0.032, 0.053]	0.026 [0.022, 0.030]

STATE: $[0, 0, 0, 0]$; A-R: $[1, 0, 0, 0]$; A-RH: $[1, 0, 1, 0]$; A-RH+YAW: $[1, 0, 1, 1]$; A-FULL: $[1, 1, 1, 1]$.

Notes. Entries are equal-context means with 95% exact cluster-bootstrap intervals over 180 rollouts per condition. A-FULL denotes A-ACTIONDIR; masks use $[r, \tau, z, \psi]$ order.

Compact channel summary. The following summary retains the main-text table's original layout. It reports terminal errors and completion rates for the same channel masks; the full means, confidence intervals, and post-entry diagnostics are given in Tables 13 and 14.

Correction-channel ablation. Table 15 extends the ablation to full trajectories. A-R raised $M_{\mathrm{rec}}^{\mathrm{joint}}$ from 38.89% to 97.22%. A-RH reduced $E_{\mathrm{final,H}}$ from 0.329 to 0.033 m, but $M_{\mathrm{lap}}^{\mathrm{joint}}$ remained

Table 14: **Post-entry geometry under correction-channel ablations.** Adding yaw to radial–height correction reduces post-entry angular error and closure distance.

Method	n_{post}	$E_{\text{maint,R}}$ (m) ↓	$E_{\text{maint,H}}$ (m) ↓	E_θ^{post} (rad) ↓	$D_{\text{close}}^{\text{post}}$ (m) ↓	$F_{\tan}$ ↑
STATE	70	2.873 [2.589,3.114]	0.216 [0.196,0.233]	4.861 [4.542,5.180]	15.699 [13.123,18.375]	0.536 [0.487,0.585]
A-R	175	0.174 [0.153,0.187]	0.314 [0.254,0.374]	1.684 [1.257,2.111]	24.799 [20.644,27.234]	0.852 [0.818,0.888]
A-RH	180	0.171 [0.159,0.183]	0.048 [0.043,0.053]	1.402 [1.053,1.755]	21.602 [17.560,24.742]	0.853 [0.827,0.886]
A-RH+YAW	180	0.050 [0.045,0.056]	0.027 [0.027,0.028]	0.207 [0.173,0.241]	5.099 [3.914,6.284]	0.967 [0.958,0.977]
A-FULL	180	0.054 [0.047,0.061]	0.028 [0.027,0.028]	0.308 [0.204,0.449]	6.522 [4.133,9.685]	0.952 [0.931,0.971]

Notes. n_{post} is a pooled count; other entries are equal-context means with 95% exact cluster-bootstrap intervals, conditioned on valid joint entry. The post superscript distinguishes these quantities from full-trajectory E_Θ and D_{close} in Table 2.

52.78%. A-RH+YAW reached 93.89%, versus 93.33% for A-FULL. Thus, adding tangential correction did not improve observed lap success. Table 14 reports the separate post-entry diagnostics.

Table 15: **Correction-channel ablation.** Adding yaw correction raises lap success from 52.78% to 93.89%.

Method	$M_{\text{rec}}^{\text{joint}}$ (%) ↑	$M_{\text{lap}}^{\text{joint}}$ (%) ↑	$E_{\text{final,R}}$ (m) ↓	$E_{\text{final,H}}$ (m) ↓
STATE	38.89	2.78	3.786	1.791
A-R	97.22	42.78	0.130	0.329
A-RH	100.00	52.78	0.115	0.033
A-RH+YAW	100.00	93.89	0.032	0.026
A-FULL	100.00	93.33	0.043	0.026

Observed effects. STATE achieved $M_{\text{rec}}^{\text{joint}} = 38.89\%$ and $M_{\text{lap}}^{\text{joint}} = 2.78\%$. A-R reached 97.22% and 42.78%, respectively, with $E_{\text{final,H}} = 0.329$ m. A-RH reduced $E_{\text{final,H}}$ to 0.033 m and had $E_{\text{maint,H}} = 0.048$ m, but $M_{\text{lap}}^{\text{joint}}$ remained 52.78%. Thus, accurate regulation of the commanded metrics did not ensure orbital progression and closure under this protocol.

Adding yaw to A-RH raised $M_{\text{lap}}^{\text{joint}}$ to 93.89%, with $E_{\text{final,R}} = 0.032$ m and $E_{\text{final,H}} = 0.026$ m. It reduced E_θ^{post} from 1.402 to 0.207 rad and $D_{\text{close}}^{\text{post}}$ from 21.602 to 5.099 m; $F_{\tan}$ increased from 0.853 to 0.967. A-FULL attained 93.33% on $M_{\text{lap}}^{\text{joint}}$, with $E_{\text{maint,R}} = 0.054$ m, $E_{\text{maint,H}} = 0.028$ m, $E_\theta^{\text{post}} = 0.308$ rad, $D_{\text{close}}^{\text{post}} = 6.522$ m, and $F_{\tan} = 0.952$. These conditional means need not satisfy the per-rollout success thresholds: the qualifying population also contains recovered rollouts that fail the full-lap criterion.

Paired comparison and scope. For $M_{\text{lap}}^{\text{joint}}$, the paired difference (A-RH+YAW minus A-FULL) was 0.56 percentage points, with a 95% exact cluster-bootstrap interval of $[-1.11, 2.22]$ percentage points. The interval includes zero: it does not establish superiority or equivalence. On this confirmation population, the radial–vertical–yaw subset retained high completion without the tangential channel. This supports a contribution from yaw in the tested subset, not a claim that every channel is necessary. Because masks are applied after training to one frozen system, the results measure sensitivity to available correction channels, not the optimality of channel-specific policies trained from scratch. A-FULL remains the unmasked configuration; retaining every channel is not itself evidence of better generalization.

H Additional Analyses

H.1 Extended experimental results and diagnostics

Additional RQ1 response statistics. On Orbit-R, OpenVLA had a 94% wrong-direction rate and a 2% near-zero rate. For π_0, wrong-direction rates were 25%, 54%, and 68% on Landing, Orbit-H, and Orbit-R, respectively; the corresponding mean responses were $+9.43$ m, -2.10 m, and -1.29 m. These statistics supplement Fig. 2 and distinguish weak response from response in the wrong direction.

Additional RQ2 comparisons. Adding the explicit residual in E reduced trajectory task-metric MAE relative to S by 4.8% on Orbit-R and 33.2% on Orbit-H. Relative to T, E increased strict ordering rates by 41.7 and 75.0 percentage points, respectively (Table 1). The main text reports the corresponding tracking improvements and the different effect of adding state alone.

RQ3 correction variation. For each development context–target group, we sampled $B = 4$ base chunks and measured the stepwise variation V_i of their bounded teacher corrections. The median V_i was 0.106 m/step for Orbit-R and 0.099 m/step for Orbit-H; every audited group had nonzero variation. Thus, different chunks required different corrections under the same pre-action condition. The scalar variation definition is given in Appendix E.4.

RQ3 analytic reference. In the short-horizon comparison of Fig. 3(b), ANALYTIC further reduced $E_{\mathrm{traj3D,R}}$ and $E_{\mathrm{traj3D,H}}$ relative to A by 0.237 m and 0.146 m, respectively. This is a privileged reference comparison, distinct from the learned-condition and matched–mismatched comparisons in the main text.

Correction-channel diagnostics. The updated confirmation protocol and E_q^{post}-based diagnostics are reported in Appendix G.4 and Tables 13 and 14. They supersede the earlier channel-mask summary; results from the separate representation ablation are retained in Appendix G.3.

Task-specific rollout readings. Figure 5 provides task-specific examples using the per-rollout labels shown in the figure. On Orbit-R, STATE had $\Theta_{\mathrm{R}} = 0.228$ rad, $D_{\mathrm{close,R}} = 34.59$ m, $E_{\mathrm{maint,R}} = 7.83$ m, and $M_{\mathrm{comp,R}} = 0$, whereas the calibrated methods completed the lap. Relative to C, A reduced $E_{\mathrm{maint,R}}$ from 0.174 to 0.067 m and $D_{\mathrm{close,R}}$ from 4.59 to 4.43 m. On Orbit-H, A reduced $E_{\mathrm{maint,H}}$ from 0.314 to 0.052 m and $D_{\mathrm{close,H}}$ from 4.92 to 4.88 m. These examples complement the aggregate results with individual tracking and trajectory geometry, rather than additional population-level estimates.

H.2 Command-response calibration

As a secondary analysis, we fit the relation between commanded and achieved metrics:

$$y_q^{\mathrm{achieved}} \approx \alpha_q y_q^{\star} + \beta_q. \tag{70}$$

Perfect calibration corresponds to $\alpha_q = 1$ and $\beta_q = 0$. We report this relation together with Pearson correlation, but do not use either statistic as a substitute for endpoint metric error or full-trajectory maintenance.

H.3 Residual and visibility breakdowns

We report performance separately for positive, negative, and zero residuals; residual-magnitude bins; task-metric bins; states in which the current target is visible or not visible; projected target-size bins; and individual anchors/landmarks. These breakdowns distinguish a uniformly useful numerical interface from one that succeeds only in restricted geometric conditions.

H.4 Tail errors and extrapolation

In addition to mean metrics, we report P50, P90, and P95 task-metric errors. Commands outside the controlled training support are evaluated only as a secondary stress test and do not alter the primary interpolation conclusions.

I SUPPLEMENTARY EXPERIMENTAL RESULTS

I.1 SINGLE-CHUNK H24 CONFIRMATION

To separate offline regulation from feedback performance, we replay one $H = 24$ chunk without re-observation (Table 16). Radius/height MAEs and expert-referenced XYZ RMSE are offline trajectory metrics, not terminal or maintenance errors. The RMSE follows Eq. equation 58.

Table 16: **Offline H24 confirmation.** A attains the lowest radius MAE, ANALYTIC the lowest height MAE, and STATE the lowest expert-referenced XYZ RMSE.

Method	Radius MAE (m) ↓	Height MAE (m) ↓	XYZ RMSE (m) ↓
State	3.381	0.558	2.564
C	1.622	0.159	4.057
A-Mismatch	1.345	0.158	4.011
A	0.961	0.083	3.789
Analytic	1.031	0.033	3.152

Notes. All errors are in meters.

I.2 ACTIONDIR H24 CONFIRMATION AND SENSITIVITY ANALYSES

This appendix reports three supplementary evaluations of the action-conditioned calibration module: calibrator capacity, inference-time correction budgets, and a single-chunk $H = 24$ confirmation replay. These experiments use the ActionDir protocol and are separate from the full-trajectory closed-loop experiments in the main paper.

All experiments use the frozen base policy, frozen calibrator checkpoints, frozen normalization statistics, and the same decoded $H = 24$ base action chunks. The action direction is inferred independently for each sampled base chunk from its signed tangential displacement. The inferred direction is used consistently for both the calibrator condition and the correction basis. Ties are resolved using the canonical counter-clockwise direction.

The experiments use seeds 33–35 and four independent physical-context clusters. Results are first averaged within each physical context and then averaged with equal weight across the four contexts. Confidence intervals are computed using an exact four-cluster bootstrap with 256 replicates.

We report the following metrics. $E_{\mathrm{traj},R}$ and $E_{\mathrm{traj},H}$ denote the trajectory-averaged radius and height MAE, respectively. $E_{\mathrm{end},R}$ and $E_{\mathrm{end},H}$ denote endpoint radius and height errors. $E^{H24}_{\mathrm{xyzRMSE}}$ denotes the expert-referenced coordinate-wise XYZ RMSE defined in Eq. equation 58. The implementation artifact retains the legacy field name `trajectory_xyz_rmse_vs_expert_m` for this metric. Each table reports the equal-weight mean followed by the 95% cluster-bootstrap confidence interval.

I.2.1 CALIBRATOR CAPACITY

We evaluate four calibrator architectures while keeping the data, base chunks, correction budget, and ActionDir direction contract fixed. The linear model has no hidden layer. The nonlinear models use two hidden layers with widths 64, 128, or 256 and SiLU activations. No model is retrained in this evaluation; the experiment replays the frozen $H = 24$ confirmation chunks under the corresponding capacity variant.

Table 17 shows that the linear model underfits the correction mapping. Relative to MLP-128, it increases the trajectory radius MAE by 2.236 m and the trajectory height MAE by 0.067 m. MLP-64 recovers most of the radius improvement but remains less accurate on height regulation. MLP-256 reduces the trajectory radius and height MAE by only 0.022 m and 0.011 m, respectively, relative to MLP-128, while increasing the parameter count from 42,976 to 118,624. These results support MLP-128 as a practical accuracy–complexity trade-off.

Paired contrast means and intervals are computed before rounding the context means; displayed contrasts may therefore differ by 0.001 from differences between the rounded table entries.

Table 17: Calibrator-capacity ablation under the ActionDir H24 confirmation protocol. Each entry is the equal-weight mean followed by the 95% exact four-cluster bootstrap interval. Lower is better for all metrics.

Calibrator	**Params.**	$E_{\mathrm{traj},R}$	$E_{\mathrm{traj},H}$	$E_{\mathrm{end},R}$	$E_{\mathrm{end},H}$	$E^{H24}_{\mathrm{xyzRMSE}}$
Linear	10,560	3.293 [3.230,3.329]	0.161 [0.158,0.164]	5.628 [5.542,5.695]	0.124 [0.122,0.126]	4.649 [4.601,4.698]
MLP-64	17,440	1.082 [1.077,1.087]	0.121 [0.120,0.122]	2.954 [2.942,2.966]	0.135 [0.133,0.138]	3.938 [3.847,4.029]
MLP-128	42,976	1.057 [1.050,1.067]	0.093 [0.092,0.094]	2.911 [2.896,2.926]	0.084 [0.082,0.086]	3.943 [3.854,4.033]
MLP-256	118,624	**1.035** **[1.025,1.047]**	**0.082** **[0.081,0.083]**	**2.853** **[2.835,2.871]**	**0.063** **[0.061,0.065]**	**3.912** **[3.821,4.004]**

Note. The bold entries identify the lowest observed value among the capacity variants for the corresponding metric. The MLP-128 checkpoint is retained as the nominal architecture because MLP-256 provides only small additional reductions in task error while requiring approximately $2.76\times$ as many parameters.

Table 18: Paired capacity contrasts relative to MLP-128. Each entry reports $\Delta = E_{\mathrm{variant}} - E_{\mathrm{MLP\text{-}128}}$ followed by the 95% exact four-cluster bootstrap interval.

Contrast	$\Delta E_{\mathrm{traj},R}$	$\Delta E_{\mathrm{traj},H}$	$\Delta E_{\mathrm{end},R}$	$\Delta E_{\mathrm{end},H}$	$\Delta E^{H24}_{\mathrm{xyzRMSE}}$
Linear − MLP-128	2.236 [2.163,2.279]	0.067 [0.064,0.070]	2.717 [2.619,2.781]	0.040 [0.037,0.042]	0.705 [0.658,0.747]
MLP-64 − MLP-128	0.025 [0.019,0.031]	0.028 [0.027,0.029]	0.042 [0.030,0.052]	0.051 [0.048,0.053]	-0.006 [-0.009,-0.002]
MLP-256 − MLP-128	-0.022 [-0.028,-0.016]	-0.011 [-0.012,-0.010]	-0.058 [-0.071,-0.046]	-0.021 [-0.024,-0.017]	-0.031 [-0.034,-0.028]

I.2.2 Correction-Budget Sensitivity

We next vary the inference-time correction limits while keeping the base policy and calibrator weights fixed. The nominal configuration is denoted by $(\alpha_{\mathrm{step}}, \alpha_{\mathrm{chunk}}) = (1.0, 1.0)$. The step and chunk limits are independently scaled by factors in $\{0.5, 1.0, 2.0\}$. This evaluation changes only inference-time clipping and does not retrain or retune the calibrator.

Table 19: Correction-budget sensitivity for A-ActionDir. The two factors scale the per-step and per-chunk correction bounds, respectively. Each entry is the equal-weight mean followed by the 95% exact four-cluster bootstrap interval.

Step / Chunk	$E_{\mathrm{traj},R}$	$E_{\mathrm{traj},H}$	$E_{\mathrm{end},R}$	$E_{\mathrm{end},H}$	$E^{H24}_{\mathrm{xyzRMSE}}$
0.5 / 0.5	2.072 [2.043,2.101]	0.346 [0.343,0.349]	3.723 [3.691,3.755]	0.216 [0.213,0.218]	3.037 [2.952,3.123]
0.5 / 1.0	2.055 [2.027,2.083]	0.347 [0.344,0.350]	3.727 [3.701,3.757]	0.221 [0.219,0.223]	3.090 [3.004,3.175]
0.5 / 2.0	2.055 [2.027,2.083]	0.347 [0.344,0.350]	3.727 [3.701,3.757]	0.221 [0.219,0.223]	3.090 [3.004,3.175]
1.0 / 0.5	1.728 [1.698,1.759]	0.134 [0.132,0.136]	3.979 [3.937,4.022]	0.358 [0.350,0.367]	3.261 [3.178,3.345]
1.0 / 1.0	1.057 [1.050,1.067]	0.093 [0.092,0.094]	2.911 [2.896,2.926]	0.084 [0.082,0.086]	3.943 [3.854,4.033]
1.0 / 2.0	1.057 [1.049,1.069]	0.093 [0.092,0.094]	2.914 [2.894,2.931]	0.084 [0.082,0.086]	3.974 [3.884,4.066]
2.0 / 0.5	1.737 [1.706,1.768]	0.136 [0.134,0.138]	4.003 [3.961,4.045]	0.376 [0.368,0.384]	3.272 [3.189,3.355]
2.0 / 1.0	1.063 [1.055,1.074]	0.091 [0.090,0.092]	2.915 [2.899,2.931]	0.083 [0.081,0.085]	3.957 [3.867,4.047]
2.0 / 2.0	1.058 [1.049,1.074]	0.091 [0.090,0.092]	2.912 [2.890,2.936]	0.083 [0.081,0.086]	4.011 [3.916,4.109]

The budget sweep reveals a clear asymmetry between the two bounds. Reducing the per-step factor to 0.5 substantially increases the task-metric error, even when the chunk-level budget is unchanged. For example, the $(1.0, 0.5)$ configuration increases the trajectory radius MAE from $1.057\,\mathrm{m}$ to $1.728\,\mathrm{m}$ and the endpoint height error from $0.084\,\mathrm{m}$ to $0.358\,\mathrm{m}$. Reducing both factors to 0.5 further increases the trajectory height MAE to $0.346\,\mathrm{m}$.

Increasing the chunk factor above the nominal value has little effect on task-metric accuracy. The $(1.0, 2.0)$ configuration changes the trajectory radius MAE by only $-0.000\,\mathrm{m}$ relative to nominal, while increasing the expert-referenced 3D error by $0.031\,\mathrm{m}$. The $(2.0, 2.0)$ configuration reduces the height MAE by only $0.002\,\mathrm{m}$ but increases the 3D error by $0.068\,\mathrm{m}$. Thus, the nominal $(1.0, 1.0)$

setting avoids the substantial under-correction caused by small step budgets without introducing the additional path deviation associated with unnecessarily large correction budgets.

The nominal-setting method comparison is reported in Table 21, which includes the same task-metric, endpoint, and expert-reference measures.

Table 20: Paired budget contrasts for A-ACTIONDIR relative to the nominal $(1.0, 1.0)$ configuration. Each entry reports $\Delta = E_{\text{variant}} - E_{\text{nominal}}$ followed by the 95% exact four-cluster bootstrap interval.

Step / Chunk	$\Delta E_{\text{traj},R}$	$\Delta E_{\text{traj},H}$	$\Delta E_{\text{end},R}$	$\Delta E_{\text{end},H}$	$\Delta E^{H24}_{\text{xyzRMSE}}$
0.5 / 0.5	1.015 [0.975,1.048]	0.253 [0.250,0.257]	0.811 [0.769,0.853]	0.132 [0.128,0.135]	-0.906 [-0.917,-0.895]
0.5 / 1.0	0.998 [0.960,1.030]	0.254 [0.250,0.257]	0.816 [0.780,0.853]	0.137 [0.134,0.140]	-0.854 [-0.865,-0.843]
0.5 / 2.0	0.998 [0.960,1.030]	0.254 [0.250,0.257]	0.816 [0.780,0.853]	0.137 [0.134,0.140]	-0.854 [-0.865,-0.843]
1.0 / 0.5	0.671 [0.631,0.705]	0.041 [0.039,0.042]	1.068 [1.014,1.107]	0.274 [0.266,0.283]	-0.682 [-0.695,-0.670]
1.0 / 2.0	-0.000 [-0.003,0.002]	0.000 [0.000,0.000]	0.002 [-0.004,0.009]	0.000 [-0.000,0.000]	0.031 [0.029,0.034]
2.0 / 0.5	0.680 [0.640,0.714]	0.043 [0.041,0.044]	1.092 [1.039,1.130]	0.292 [0.284,0.300]	-0.671 [-0.685,-0.659]
2.0 / 1.0	0.006 [0.005,0.007]	-0.002 [-0.003,-0.002]	0.004 [0.002,0.005]	-0.001 [-0.001,-0.001]	0.013 [0.012,0.015]
2.0 / 2.0	0.001 [-0.003,0.007]	-0.002 [-0.003,-0.002]	0.001 [-0.008,0.011]	-0.001 [-0.001,-0.000]	0.068 [0.060,0.077]

Paired contrast means and intervals are computed before rounding the context means; displayed contrasts may therefore differ by 0.001 from differences between the rounded table entries.

I.2.3 SINGLE-CHUNK H24 CONFIRMATION UNDER ACTIONDIR

The following experiment evaluates a single decoded $H = 24$ action chunk under world-frame geometric replay. It is a landmark-disjoint confirmation evaluation and does not perform replanning after executing the chunk. Therefore, the results below should not be interpreted as full-trajectory closed-loop or physical-dynamics evidence.

The evaluated methods are STATE, C-ACTIONDIR, A-ACTIONDIR, A-M-ACTIONDIR, and the analytic geometric reference. For A-ACTIONDIR, the direction condition is inferred from the corresponding corrected base chunk. For A-M-ACTIONDIR, the calibrator receives the action summary from a same-state peer chunk rather than the chunk being corrected. The latter provides the matched–mismatched intervention while keeping the task condition and physical context fixed.

Table 21: Single-chunk $H = 24$ confirmation results under the ActionDir protocol. Each entry is the equal-weight mean followed by the 95% exact four-cluster bootstrap interval. Bold entries identify the best learned method for task-metric errors; the analytic reference is shown separately.

Method	$E_{\text{traj},R}$	$E_{\text{traj},H}$	$E_{\text{end},R}$	$E_{\text{end},H}$	E^{H24}_{xyzRMSE}
STATE	3.381 [3.329,3.431]	0.558 [0.552,0.563]	4.783 [4.739,4.827]	0.149 [0.144,0.153]	**2.564** **[2.489,2.640]**
C-ACTIONDIR	1.933 [1.892,1.988]	0.199 [0.190,0.208]	3.981 [3.916,4.053]	0.163 [0.162,0.165]	4.146 [4.083,4.223]
A-M-ACTIONDIR	1.455 [1.399,1.517]	0.167 [0.161,0.172]	3.617 [3.497,3.739]	0.198 [0.192,0.203]	4.143 [4.020,4.265]
A-ACTIONDIR	**1.057** **[1.050,1.067]**	**0.093** **[0.092,0.094]**	**2.911** **[2.896,2.926]**	**0.084** **[0.082,0.086]**	3.943 [3.854,4.033]
ANALYTIC	1.031 [1.005,1.053]	0.033 [0.032,0.033]	3.046 [2.955,3.117]	0.000 [0.000,0.000]	3.152 [3.131,3.172]

The single-chunk confirmation reproduces the main ordering among learned methods. A-ACTIONDIR reduces the trajectory radius MAE by $0.876\,\text{m}$ relative to C-ACTIONDIR, and by $0.398\,\text{m}$ relative to A-M-ACTIONDIR. The corresponding reductions in trajectory height MAE are $0.106\,\text{m}$ and $0.074\,\text{m}$, respectively. The same ordering is observed at the endpoint: A-ACTIONDIR reduces endpoint radius error by $1.069\,\text{m}$ relative to C-ACTIONDIR and by $0.706\,\text{m}$ relative to A-M-ACTIONDIR.

The comparison with STATE is task-dependent. Relative to STATE, A-ACTIONDIR reduces trajectory radius MAE by $2.324\,\text{m}$ and trajectory height MAE by $0.465\,\text{m}$. However, STATE has

lower expert-referenced 3D trajectory error. This difference is expected from the metric definitions: expert-path agreement and commanded radius–height regulation measure distinct properties and should not be treated as interchangeable objectives.

The analytic reference remains strongest on height regulation and has a slightly lower trajectory radius MAE than A-ACTIONDIR. In contrast, A-ACTIONDIR has a lower endpoint radius error than the analytic reference. These results therefore support the use of action-conditioned calibration for numerical regulation, while retaining the analytic controller as a geometric reference rather than claiming uniform superiority of the learned calibrator.

Table 22: Paired contrasts for the single-chunk $H = 24$ confirmation. Each entry reports $\Delta = E_{\text{A-ActionDir}} - E_{\text{comparison}}$ followed by the 95% exact four-cluster bootstrap interval.

Comparison	$\Delta E_{\text{traj},R}$	$\Delta E_{\text{traj},H}$	$\Delta E_{\text{end},R}$	$\Delta E_{\text{end},H}$	$\Delta E^{H24}_{\text{xyzRMSE}}$
C-ACTIONDIR	-0.876 [-0.921,-0.837]	-0.106 [-0.115,-0.097]	-1.069 [-1.135,-0.997]	-0.079 [-0.081,-0.078]	-0.203 [-0.232,-0.177]
A-M-ACTIONDIR	-0.398 [-0.451,-0.346]	-0.074 [-0.078,-0.069]	-0.706 [-0.829,-0.582]	-0.114 [-0.120,-0.110]	-0.199 [-0.233,-0.166]
STATE	-2.324 [-2.378,-2.263]	-0.465 [-0.470,-0.459]	-1.872 [-1.912,-1.817]	-0.065 [-0.069,-0.062]	1.379 [1.353,1.408]
ANALYTIC	0.026 [0.002,0.061]	0.061 [0.059,0.062]	-0.134 [-0.202,-0.032]	0.084 [0.082,0.086]	0.791 [0.723,0.869]

Paired contrast means and intervals are computed before rounding the context means; displayed contrasts may therefore differ by 0.001 from differences between the rounded table entries.

I.2.4 SUMMARY OF SUPPLEMENTARY FINDINGS

The three evaluations provide complementary evidence for the ActionDir design. First, calibrator capacity matters: a linear correction model is insufficient, whereas MLP-128 captures most of the available nonlinear correction benefit without the parameter cost of MLP-256. Second, correction budgets are not arbitrary implementation details. Small per-step budgets substantially limit numerical recovery, while larger budgets provide negligible task-metric gains and can increase deviation from the expert-referenced trajectory. Third, the single-chunk $H = 24$ confirmation shows that the matched A-ACTIONDIR input is consistently more accurate than both the condition-only C-ACTIONDIR model and the same-state mismatched A-M-ACTIONDIR intervention on the commanded radius and height metrics.

These experiments are limited to frozen decoded chunks and additive world-frame geometric replay. They do not evaluate inertial dynamics, actuator limits, observation latency, physical disturbances, or real-flight execution. Accordingly, the results support sensitivity and mechanism claims within the H24 replay protocol and should not be interpreted as independent evidence for physical-dynamics robustness.

I.3 EXECUTION-MISMATCH CONFIRMATION

The mismatch study evaluates 11 perturbations with 180 rollouts per method and condition. All 33 audits passed and all 11 aggregates were complete, with 900 records each. Table 23 reports success and A's terminal absolute errors, not maintenance MAEs. A-Mismatch has higher success than A in nine conditions; composite mismatch raises A's terminal errors to 0.490/0.245 m. These outcomes do not support numerical insensitivity to mismatch. This perturbation matrix is separate from the nominal Analytic-ActionDir row in Table 2.

Table 23: **Full-lap completion under execution mismatch.** A exceeds STATE and C in every condition, while A-MISMATCH achieves higher success in nine conditions and composite mismatch yields the largest terminal errors for A.

Condition	$M_{\text{lap}}^{\text{joint}}$ (%) ↑					A: final MAE (m) ↓	
	State	C	A-M	A	Analytic-ActionDir	Radius error	Height error
Action scale 0.8	1.67	83.33	91.11	88.33	47.78	0.096	0.025
Action scale 0.9	1.67	87.22	93.89	92.78	42.78	0.057	0.024
Action scale 1.1	1.11	68.33	97.78	93.33	45.56	0.063	0.025
Action scale 1.2	2.22	57.22	94.44	96.67	43.33	0.086	0.027
One-step delay	2.22	85.00	93.33	91.11	50.00	0.259	0.027
Persistent local wind bias	0.56	81.67	95.00	91.67	45.56	0.179	0.094
State noise	2.22	80.56	92.78	92.22	47.78	0.090	0.071
Reference-coordinate noise	2.78	81.67	95.00	92.78	43.89	0.232	0.168
Actuator limits	2.78	82.78	93.33	92.78	77.78	0.044	0.023
Yaw–translation coupling	2.22	77.22	93.89	91.11	46.11	0.067	0.025
Composite mismatch	0.56	73.89	90.56	91.11	48.33	0.490	0.245

Notes. Each method has 180 rollouts per condition. Method columns report joint full-lap success (%); the last two columns report only A's terminal absolute errors (m), not maintenance MAEs.