# Turning VLAs into Interactive Tools for Agentic Robotic Manipulation
# Supplementary Material

Table S1. **Notation.** All geometric quantities use the adapter's declared robot-base frame.

| Symbol | Definition |
|---|---|
| $x_t$, $v_\theta$ | Noisy action chunk and action-expert velocity |
| $c_k$, $m_t$ | Cached conditioning and clean-action estimate |
| $\mathcal{C}_{o_k}$ | Differentiable native-command-to-pose map |
| $E_h$, $q_h(t)$ | Complete-pose error and arrival assignment |
| $M$, $M_{e,c}$ | Valid/targeted action mask and pose-component mask |
| $\mathcal{B}_t$, $C_t$ | Bounded guidance and target compatibility |
| $H$, $H_{\text{exec}}$ | Generated horizon and executed prefix length |
| $k$, $h$, $t$ | Decision, action, and generation-time indices |

## A. Implementation Details and Differentiable Guidance

**Scope and notation.** The following reference configuration specifies the interface, numerical operations, and execution rules needed to implement InteractVLA. We distinguish the environment decision $k$, the action index $h \in \{1, \dots, H\}$, and the generation time $t \in [0, 1]$. Given observation $o_k$, instruction $\ell$, and conditioning $c_k$, the policy generates $A_k \in \mathbb{R}^{H \times D}$; execution consumes a prefix of length $H_{\text{exec}}$. An agent-specified target $G_k = (p^*, R^*, w^*)$ contains position, orientation, and commanded gripper width. Effector and component masks select the fields constrained by the call. Generation time parametrizes denoising, while action indices identify subsequent controller commands.

### A.1. Native Actions and Command-to-Pose Mappings

**Adapter contract.** Each checkpoint is accompanied by a manifest containing action-channel order, valid width, padding, normalization statistics, rotation convention, controller scale and saturation, gripper polarity, joint ordering, frame transforms, and controller rate. The reference adapters in Table S2 use these entries to construct $\mathcal{C}_{o_k}$. Initialization checks their shapes and semantics against the checkpoint and controller; an inconsistent manifest terminates initialization with an explicit configuration error. The dimensionalities in the table specify the reference adapter layouts. The loaded manifest determines the valid output width, including any padded channels.

**Inverse normalization.** For a joint channel normalized to $[-1, 1]$, decoding uses the checkpoint's lower and upper quantiles $l_d, u_d$:

$$a_d^{\text{phys}} = l_d + \frac{\text{clip}(a_d, -1, 1) + 1}{2}(u_d - l_d). \tag{A.1}$$

Cartesian adapters apply the checkpoint-specified inverse transform before controller scaling. A mean/standard-deviation transform gives $a_d^{\text{phys}} = \sigma_d a_d + \mu_d$; a quantile transform uses Eq. (A.1). The manifest selects exactly one transform per channel. Padding is excluded from decoding, gradients, and norm statistics. Controller saturation is included in the geometric branch using the same bounds as execution; its derivative is one in the interior and zero on saturated coordinates.

**Cartesian commands.** An absolute-pose adapter extracts the decoded position and rotation and transforms them into the common base frame. An incremental adapter composes rotations on $SO(3)$ using the controller's convention. For the world-frame delta operational-space reference adapter, the accumulated command state obeys

$$\bar{p}_1 = p_{\text{now}}, \qquad \bar{R}_1 = R_{\text{now}}, \tag{A.2}$$

$$p_h^{\text{cmd}} = \bar{p}_h + \Delta p_h, \qquad R_h^{\text{cmd}} = \text{Exp}([\Delta r_h]_\times)\bar{R}_h, \tag{A.3}$$

$$\bar{p}_{h+1} = \bar{p}_h + \eta \Delta p_h, \qquad \bar{R}_{h+1} = \text{Exp}([\eta \Delta r_h]_\times)\bar{R}_h. \tag{A.4}$$

Here $[\cdot]_\times$ maps a vector to its skew-symmetric matrix, and the reference tracking coefficient is $\eta = 0.25$. A body-frame adapter instead right-multiplies rotations and rotates translational increments by the accumulated orientation. Selecting this convention is part of loading the controller. All rotations are composed as matrices; rotation-vector and quaternion interfaces are converted before composition. Quaternions are normalized before matrix conversion, which gives identical errors for $q$ and $-q$.

**Joint commands.** For arm $e$ with decoded joint positions $q_{h,e}$, the reference joint adapter evaluates differentiable forward kinematics,

$${}^bT_{h,e} = {}^bT_{e,0} \prod_{j=1}^{n_e} T_{e,j}(q_{h,e,j}), \qquad (p_{h,e}, R_{h,e}) = \text{pose}({}^bT_{h,e}). \tag{A.5}$$

The ordered joints, link transforms, and joint limits come from the same robot model used by the controller. The fixed mount transform ${}^bT_{e,0}$ expresses both arms in a common base frame. Command geometry and measured geometry are logged separately: the former defines the differentiable objective, and the latter records the achieved configuration after execution.

**Gripper semantics.** The tool accepts a normalized target gripper value $g_e^* \in [0, 1]$, with zero denoting closed and one fully open. The adapter converts it to the internal width

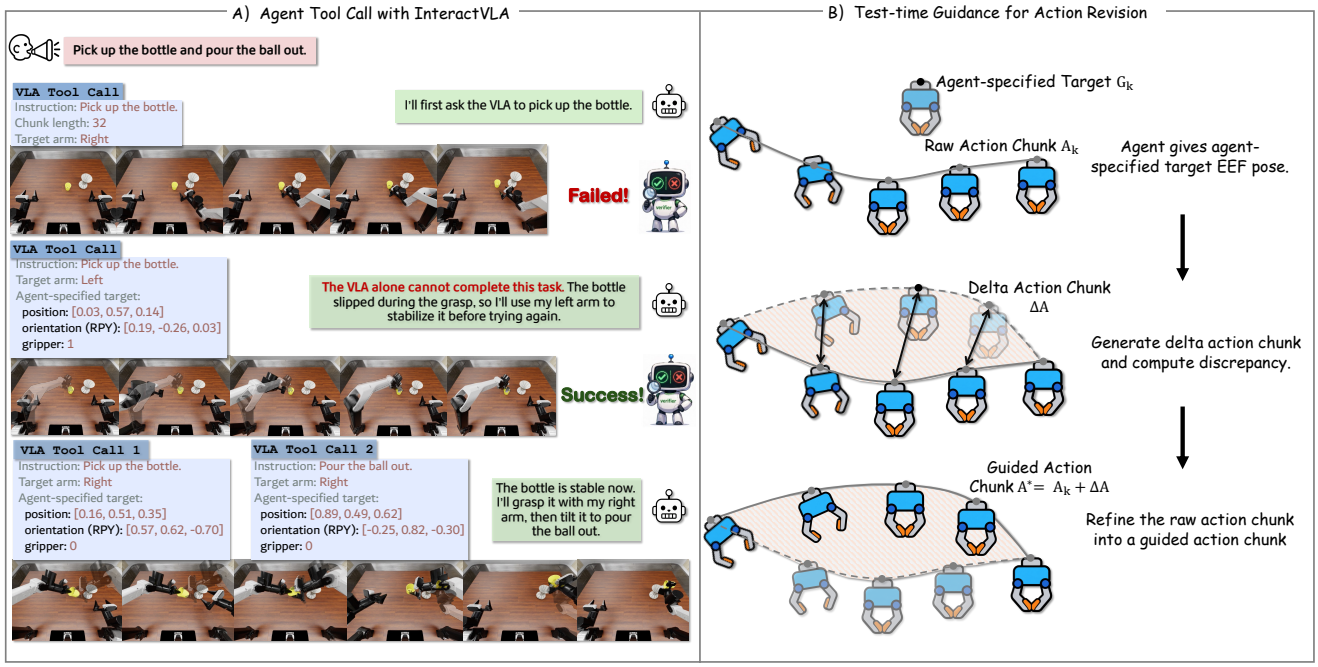



Figure 1. **InteractVLA turns VLAs into interactive tools for agents.** *Left*: When a VLA call fails, the agent supplies spatial targets to steer subsequent motions toward its intended plan. *Right*: InteractVLA uses the agent-specified target to guide the denoising direction during action generation, refining a raw action chunk into a guided action chunk.

target $w_e^* = w_{\min,e} + g_e^*(w_{\max,e} - w_{\min,e})$ in metres. A continuous gripper command is mapped by the controller's affine width calibration, $w = a_e q^{\text{grip}} + b_e$, including its physical limits. The historical RoboDojo command-width convention corresponds to $a_e = 0.108$ and $b_e = -0.020$; the reference adapter stores these coefficients as controller metadata. Binary-action controllers use a differentiable distribution over nine states $s_j = -1 + j/4$, $j = 0, \ldots, 8$, where increasing $s_j$ closes the gripper. Their associated widths are $w_j = w_{\min} + (w_{\max} - w_{\min})(1 - s_j)/2$. At index $h$, the closing probability is

$$P_h^{\text{close}} = \Phi\left(a_h^{\text{grip}} \frac{t}{1-t}\right). \tag{A.6}$$

The state distribution advances one bin toward closed or open with these probabilities, saturating at its endpoints. Its initial distribution linearly interpolates the two bins surrounding the measured width. The loss uses $\sum_j \pi_{h,j}(w_j - w^*)^2$. Execution applies the native sign threshold at zero; the manifest maps this internal closing convention to the controller's polarity. The binary limit supplies endpoint diagnostics.

**Bimanual targets.** The channel order is left arm then left gripper, followed by right arm then right gripper. Pose masks $M_{e,c} \in \{0, 1\}$ select position, rotation, and gripper constraints. The action mask $M$ also removes non-targeted-arm channels from the applied correction; their state updates retain the native velocity. A target for simultaneous two-arm coordination sums both arms' errors before temporal alignment. Sequential coordination uses successive calls: for example, a left-arm stabilization target is executed and verified before a right-arm manipulation target is issued. An explicit hold constraint can keep the stabilizing arm near its measured pose during the second call.

## A.2. Complete-Pose Objective

For effector $e$, the complete-pose discrepancy is

$$E_h = \sum_e \sum_{c \in \{p,R,w\}} M_{e,c} \lambda_{e,c} \rho_{h,e,c}, \tag{A.7}$$

$$\rho_{h,e,p} = \frac{\|p_{h,e} - p_e^*\|_2^2}{s_p^2}, \tag{A.8}$$

$$\rho_{h,e,R} = \frac{\|R_{h,e} - R_e^*\|_F^2}{s_R^2}, \tag{A.9}$$

$$\rho_{h,e,w} = \frac{\mathbb{E}[(w_{h,e} - w_e^*)^2]}{s_{w,e}^2}. \tag{A.10}$$

The expectation is a point mass for continuous width commands and the nine-state distribution for a binary controller. We use $s_p = 0.05$ m, $s_R = 2\sqrt{2}\sin(15°)$, $s_{w,e} = w_{\max,e} - w_{\min,e}$, and $\lambda_{e,c} = 1$. For relative rotation angle $\theta$, $\|R - R^*\|_F^2 = 8\sin^2(\theta/2)$, so a 30° discrepancy gives a unit rotation error. These choices make a component's unit error correspond to its declared geometric scale.

Table S2. **Reference controller adapters.** Joint formats use differentiable forward kinematics; delta formats use Eq. (A.4). Checkpoint metadata supplies channel ordering, normalization statistics, controller bounds, and gripper calibration. Adapter initialization checks the declared action format and control rate.

| Platform / backbone | Native reference layout | Frame / units | Inverse normalization | Pose mapping | $H/H_{\text{exec}}$ | Rate |
|---|---|---|---|---|---|---|
| LIBERO-Plus / MR-0 | Delta pose + gripper (7) | Base; m, rad | Checkpoint transform | Delta command tracking | 10/10 | 20 Hz |
| RoboDojo / MR-0 | Two joint arms + grippers (14) | Base; rad, m | Checkpoint quantiles | Dual-arm FK | 50/16 | 20 Hz |
| RoboCasa365 / MR-0 | Delta pose + gripper (7) | Base; m, rad | Checkpoint transform | Delta command tracking | 50/16 | 20 Hz |
| Dexmal / MR-0 | Two joint arms + grippers (14) | Base; rad, m | Checkpoint quantiles | Dual-arm FK | 32/8 | 10 Hz |
| RoboDojo / $\pi_{0.5}$ | Two joint arms + grippers (14) | Base; rad, m | Checkpoint quantiles | Dual-arm FK | 50/16 | 20 Hz |

All active fields are aggregated before assigning an arrival index. This represents the agent's complete configuration at a common point in the chunk. For two constrained effectors, that same index is shared across both arms. The objective is evaluated in float32, including rotation conversion and width moments; stable small-angle expansions implement the exponential map. Invalid rotations or zero-norm input quaternions return an invalid-target error before generation.

## A.3. Gradient Computation

The clean estimate and command-pose sequence at generation time $t$ are

$$m_t = x_t + (1-t)v_\theta(x_t, t; c_k), \qquad \mathcal{P}_t = \mathcal{C}_{o_k}(m_t). \quad \text{(A.11)}$$

For the objective $L_t(\mathcal{P}_t, G_k)$, the input gradient is

$$g_t = [I + (1-t)J_{v_\theta}(x_t)]^\top J_{\mathcal{C}_{o_k}}(m_t)^\top \nabla_{\mathcal{P}_t} L_t. \quad \text{(A.12)}$$

The action expert and accumulated-command mapping can couple multiple action indices. Autodifferentiation retains this current-state dependence, including the action expert's attention across the chunk. After each integration step the new state is detached, so each backward pass differentiates the current expert evaluation. Model parameters have gradient accumulation disabled. Vision-language features and prefix key/value tensors are computed once per observation and cached as constants; a new observation rebuilds this conditioning.

The reference solver uses ten explicit Euler evaluations at $t_i = i/10$, $i = 0, \ldots, 9$. Expert matrix operations may use bfloat16, while the geometric branch, reductions, norms, and state update use float32. The action input is created with gradient tracking enabled inside the guided window; inference-mode contexts are exited before these evaluations. Gradient checkpointing is disabled in the reference configuration. A nonfinite pose, loss, or gradient terminates the proposal with a numerical-error record containing the failing operation and generation index. This record is returned before any proposed command is executed.

**Native sampler conventions.** The adapter converts the checkpoint's flow convention to increasing canonical time. A velocity head using decreasing $\tau = 1 - t$ gives $m = x_\tau - \tau v_{\text{native}}$ and

$$x_{\tau+\Delta\tau} = \text{sg}\big[x_\tau + \Delta\tau\big(v_{\text{native}} + \mathcal{B}_t(g_t)\big)\big], \quad \Delta\tau < 0. \quad \text{(A.13)}$$

A clean-action head gives $v_\theta = (m_t - x_t)/(1-t)$ at the interior evaluation points. The final Euler state is decoded directly. This conversion aligns guidance signs and clean estimates across the two parameterizations.

## A.4. Guidance Operator and Hyperparameters

Let $M_{h,d}$ select valid, actively steered action coordinates and $n_M = \sum_{h,d} M_{h,d} > 0$. Define

$$\text{RMS}_M(z) = \sqrt{\frac{1}{n_M}\sum_{h,d} M_{h,d} z_{h,d}^2}, \quad \text{(A.14)}$$

$$\gamma_t = \min\left\{\lambda, \frac{\rho\,\text{RMS}_M(v_\theta)}{\text{RMS}_M(g_t) + \epsilon}\right\}, \quad \text{(A.15)}$$

$$\mathcal{B}_t(g_t) = \mathbf{1}_{\{t \in \mathcal{W}\}}\gamma_t(M \odot g_t). \quad \text{(A.16)}$$

The same mask is used in numerator and denominator, and the ratio is taken over the whole active chunk. The update is

$$x_{t+\Delta t} = \text{sg}\big[x_t + \Delta t\big(v_\theta - \mathcal{B}_t(g_t)\big)\big]. \quad \text{(A.17)}$$

When native velocity vanishes, the correction budget vanishes continuously. The reference values are $\lambda = 1$, $\rho = 0.3$, $\epsilon = 10^{-6}$, and $\mathcal{W} = [0.5, 0.8)$. Thus guided evaluations occur at $t = 0.5, 0.6, 0.7$; the remaining evaluations follow the native field. The decoded controller's saturation supplies the final action bounds. No additional smoothing is applied.

The compatibility score uses the same current gradient and pose errors,

$$C_t = \exp\left[-\max\left\{\frac{\min_h E_h}{r_0}, \frac{\lambda\,\text{RMS}_M(g_t)}{d_0(\text{RMS}_M(v_\theta) + \epsilon_v)}\right\}\right], \quad \text{(A.18)}$$

with $r_0 = 1$, $d_0 = 3$, $\epsilon_v = 10^{-3}$, and threshold $\kappa = e^{-1}$. The first active update at $t = 0.5$ is applied before checking compatibility. Checks then occur at $t = 0.6$ and $0.7$ before their updates; equality with the threshold accepts the proposal. Detailed properties and calibration appear in Sec. E.

Table S3. **Reference configuration.** Geometry, compatibility, and solver settings are shared across platforms. Development tasks select configuration choices; evaluation runs retain the chosen configuration. Episode budgets include all controller commands and agent calls within one attempt.

| Parameter | LIBERO-Plus | RoboDojo | RoboCasa365 | Dexmal | Selection / rule |
|---|---|---|---|---|---|
| Solver / evaluations | Euler / 10 | Euler / 10 | Euler / 10 | Euler / 10 | Fixed reference solver |
| Chunk / executed prefix | 10/10 | 50/16 | 50/16 | 32/8 | Adapter/controller interface |
| Gain / relative cap | 1/0.3 | 1/0.3 | 1/0.3 | 1/0.3 | Shared development configuration |
| Active window | [.5, .8) | [.5, .8) | [.5, .8) | [.5, .8) | Three active evaluations |
| Position scale / weight | .05 m / 1 | .05 m / 1 | .05 m / 1 | .05 m / 1 | Geometric tolerance |
| Rotation scale / weight | $2\sqrt{2}\sin(15°)$ / 1 | | | | 30° unit discrepancy |
| Gripper scale / weight | Controller width range / 1 | | | | Controller calibration |
| Compatibility $(r_0, d_0, \kappa)$ | $(1, 3, e^{-1})$ | | | | Shared reference boundary |
| Compatibility checkpoints | .6, .7 | .6, .7 | .6, .7 | .6, .7 | After first guided update |
| Controller-step budget | 1,000 | 2,000 | 2,000 | Time limited | Per-attempt termination |
| Agent-call budget | 64 | 128 | 128 | 128 | Per-attempt termination |
| Execution duration limit | 50 s | 100 s | 100 s | 600 s | Simulated time / Dexmal wall time |
| Replan / terminal rejections | 3/6 | 3/6 | 3/6 | 3/6 | Accepted generation resets count |

### A.5. End-to-End Execution Procedure

The interface separates proposal construction from execution. Tool fields and permissions appear in Table S8.

1. Acquire a synchronized observation and robot state, assign an observation version, and validate the target, masks, and coordinate frame. Cache conditioning and initialize the native noise sample with the recorded seed.
2. Run the ten-step sampler. At active steps construct command poses, compute Eq. (A.7), differentiate soft-time alignment, and apply Eq. (A.16). Perform the two compatibility checks on schedule.
3. Return an accepted proposal identifier, full action array, pose sequence, confidence diagnostics, and visual feedback. A rejected proposal is returned for inspection with a reason code and has execution disabled.
4. The agent inspects the proposal and explicitly requests execution of at most $H_{\text{exec}}$ commands. Execution requires the proposal's observation/state version to match the current version. Record actual commands, execution status, timestamps, and measured states.
5. Refresh observations after the prefix. The agent may regenerate with a revised target, continue the task with a fresh proposal, or request a bounded local edit using the supplied motion as a reference. An unused continuation is rebased to the current state before editing and receives a new proposal identifier.

Successful task verification, episode termination, exhausted call/step/time budgets, and controller faults end the attempt. Three consecutive target rejections trigger a fresh observation and task-level replanning; six consecutive rejections end the attempt. An accepted generation resets the counter. A tool timeout returns an error status and invalidates its pending proposal. Further actions require a fresh observation and explicit call. Targets for a subsequent task stage are issued only after the current stage's observed completion predicate is satisfied.

## B. Soft-Time Target Alignment: Derivation and Execution Semantics

### B.1. Entropy-Regularized Arrival-Time Assignment

Let $\tau = 1 - t$, $\beta_t = t^2/\tau^2$, and $u_h = 1/H$. For finite pose errors and $0 < t < 1$, define

$$F_t(E) = \min_{q \in \Delta_H} \left[ \sum_h q_h E_h + \frac{1}{\beta_t} \mathrm{KL}(q \| u) \right], \qquad \text{(B.1)}$$

where $\Delta_H$ is the probability simplex and $0 \log 0 = 0$.

**Proposition 1 (arrival assignment).** The unique minimizer and minimum in Eq. (B.1) are

$$q_h^*(t) = \frac{e^{-\beta_t E_h}}{Z_t}, \quad Z_t = \sum_j e^{-\beta_t E_j}, \qquad \text{(B.2)}$$

$$F_t(E) = -\frac{1}{\beta_t} \log(Z_t/H). \qquad \text{(B.3)}$$

*Proof.* For $q_h > 0$, stationarity of the Lagrangian gives $E_h + \beta_t^{-1}(\log(q_h/u_h) + 1) + \mu = 0$. Normalization yields Eq. (B.2), which is positive for every finite $E_h$. More directly, any $q \in \Delta_H$ satisfies

$$\sum_h q_h E_h + \beta_t^{-1} \mathrm{KL}(q \| u) = -\beta_t^{-1} \log(Z_t/H) + \beta_t^{-1} \mathrm{KL}(q \| q^*). \qquad \text{(B.4)}$$

Nonnegativity of KL divergence gives the minimum, and equality requires $q = q^*$. □

The main soft-time objective and the guidance objective are

$$L_{\text{soft}}(t) = t^2 F_t(E) = -\tau^2 \log(Z_t/H), \tag{B.5}$$

$$L_{\text{guide}}(t) = \frac{\tau}{t} L_{\text{soft}}(t) = t\tau F_t(E). \tag{B.6}$$

At fixed generation time, $\partial F_t/\partial E_h = q_h^*$. Consequently,

$$g_t = \nabla_{x_t} L_{\text{guide}}(t) = t\tau \sum_h q_h^*(t) \nabla_{x_t} E_h. \tag{B.7}$$

This is the derivative of the log-sum-exp objective, including the optimization of temporal assignment. The weighted sum in Eq. (B.7) is its closed-form gradient and is used directly by autodifferentiation of Eq. (B.6).

## B.2. Limits, Scaling, and Numerical Stability

**Proposition 2 (bounds and concentration).** Writing $E_{\min} = \min_h E_h$, the soft minimum satisfies

$$E_{\min} \le F_t(E) \le E_{\min} + \frac{\log H}{\beta_t}. \tag{B.8}$$

At fixed $E$, $q^* \to u$ as $\beta_t \to 0$; as $\beta_t \to \infty$, mass is uniform over the minimum-error indices. If the minimizer $h^*$ is unique and its error gap is $\Delta = \min_{j \neq h^*}(E_j - E_{h^*}) > 0$, then

$$q_{h^*}^* \ge \frac{1}{1 + (H-1)e^{-\beta_t \Delta}}. \tag{B.9}$$

*Proof.* Factor $e^{-\beta_t E_{\min}}$ out of $Z_t$. The remaining sum lies in $[1, H]$, yielding Eq. (B.8). The limits follow termwise from the exponential weights. Bounding each nonminimum term by $e^{-\beta_t \Delta}$ gives the final inequality. □

**Proposition 3 (sensitivity).** At fixed $\beta > 0$,

$$\frac{\partial q_i}{\partial E_j} = -\beta q_i(\mathbf{1}_{i=j} - q_j). \tag{B.10}$$

For an error perturbation with $\|\delta E\|_\infty \le \varepsilon$, integration of this Jacobian along the line segment from $E$ to $E + \delta E$ gives

$$\|q(E + \delta E) - q(E)\|_1 \le \min(2, 2\beta\varepsilon). \tag{B.11}$$

Indeed, $\sum_i q_i |\delta E_i - \sum_j q_j \delta E_j| \le 2\varepsilon$. The soft value is also 1-Lipschitz in the infinity norm because its gradient is a probability vector. These bounds quantify how concentrated assignments become more sensitive to small differences between similar arrival candidates.

For fixed errors, temporal entropy $\mathcal{H}(q) = -\sum_h q_h \log q_h$ obeys

$$\frac{d\mathcal{H}}{d\beta} = -\beta \operatorname{Var}_q(E) \le 0. \tag{B.12}$$

To see this, use $\mathcal{H} = \log Z + \beta \mathbb{E}_q E$ and $d\mathbb{E}_q E/d\beta = -\operatorname{Var}_q(E)$. During denoising, the error vector also changes with $x_t$. Its observed entropy therefore records the joint effect of this motion and the temperature schedule. Figure S2 separates these two effects in an analytical example.

**Shared complete-pose timing.** Aggregation before temporal assignment preserves configuration consistency. For example, a two-step chunk with component errors $(E_{1,p}, E_{1,R}) = (0, 1)$ and $(E_{2,p}, E_{2,R}) = (1, 0)$ has $\min_h(E_{h,p} + E_{h,R}) = 1$, whereas $\min_h E_{h,p} + \min_h E_{h,R} = 0$. The shared objective identifies the remaining full-pose mismatch. As its assignment concentrates, the direct loss contribution focuses on the matching action index; Eq. (A.12) distributes the resulting input correction through the proposal's coupled structure.

**Stable evaluation.** The reference implementation computes

$$F_t = E_{\min} - \frac{1}{\beta_t}\left[\text{logsumexp}_h\{-\beta_t(E_h - E_{\min})\} - \log H\right] \tag{B.13}$$

and obtains $q_h$ from the same shifted logits. All exponents are nonpositive. The active window keeps $\beta_t$ finite and bounded away from zero; the sampler sets guidance to zero outside this window. Endpoint diagnostics use uniform assignment at $t = 0$ and the hard-minimum limit at $t = 1$. Spatial scales in Eq. (A.10) control the relative contribution of physical quantities; $\beta_t$ controls temporal concentration; and $t(1-t)$ controls the nominal gradient envelope. The final applied magnitude also depends on the relative cap.

**Proposition 4 (local descent relative to the native step).** Fix $t$, conditioning, and a differentiable guidance objective $f(x) = L_{\text{guide}}(x, t)$. Vectorize the action chunk; norms in this proposition are Euclidean, and $M$ denotes the diagonal projector corresponding to the action mask. Suppose the gradient of $f$ is $L$-Lipschitz on a convex region containing the relevant steps. Let $g = \nabla f(x)$ and let $v$ be the native field at $x$. Compare the native update $z = x + \delta v$ with $y = z - \delta\gamma Mg$, where $\delta, \gamma > 0$. If $\|Mg\| > 0$, then

$$f(y) - f(z) \le -\delta\gamma\|Mg\|^2 + L\delta^2\gamma\|v\|\|Mg\| + \tfrac{1}{2}L\delta^2\gamma^2\|Mg\|^2. \tag{B.14}$$

In particular, $f(y) < f(z)$ whenever

$$L\delta\left(\frac{\|v\|}{\|Mg\|} + \frac{\gamma}{2}\right) < 1. \tag{B.15}$$

*Proof.* Smoothness around $z$ gives $f(y) \le f(z) - \delta\gamma\nabla f(z)^\top Mg + \frac{1}{2}L\delta^2\gamma^2\|Mg\|^2$. Because $g^\top Mg = \|Mg\|^2$ and $\|\nabla f(z) - g\| \le L\delta\|v\|$, Cauchy–Schwarz yields Eq. (B.14). □

This proposition describes the fixed-time command objective in a smooth local region, including the current expert Jacobian. The bound explicitly shows the roles of Euler step size, native motion, and correction magnitude. For piecewise-smooth clipping, it applies within a region

with fixed saturation pattern. The relative cap separately supplies the exact sampling-space bound $\mathrm{RMS}_M(\mathcal{B}_t) \le \rho\,\mathrm{RMS}_M(v)$.

### B.3. Arrival Time Under Receding-Horizon Execution

The arrival index describes a configuration within a proposed chunk. Its relation to execution is tracked explicitly. A task-stage target retains its identity across prefixes, while each geometric instance is recomputed from the newest observation and attached to that observation's version. If $h^* > H_{\mathrm{exec}}$, the prefix is treated as an approach segment and the next call uses the measured state to regenerate the continuation. For example, a RoboDojo proposal may match the target at index 35 of its 50 actions while executing the first 16 as its current approach. The next proposal is grounded in the resulting measured state. The counterfactual gate diagnostic in Sec. E executes the complete ungated proposal in an offline simulator copy to evaluate the target's compatibility with that generated chunk; the deployed agent retains prefix execution and observation-driven replanning.

The reference progress monitor evaluates the same scaled position and rotation discrepancies on measured poses, and the gripper term on the measured commanded width. Geometric arrival requires every active component to have normalized error at most one. Stage completion additionally requires the agent's observed task predicate, such as a stable grasp or the object reaching its destination. Over two executed prefixes, a target is revised if the best measured aggregate error has fallen by less than 20%. This event initiates new target selection or a local correction and prevents indefinite renewal of a distant arrival index. Episode budgets remain active throughout.

A target outside the current horizon can be approached through compatible intermediate targets. The agent retains the stage goal and selects an intermediate configuration within the current proposal's reachable neighborhood. Targets rejected by the compatibility rule return their diagnostic residuals for revision. After every executed prefix, dynamic-scene targets are regrounded using the newest RGB-D and proprioception. A stage change invalidates older proposals. Simultaneous bimanual constraints share one arrival index; sequential stages verify the first arm's condition before issuing the second-arm target.

### B.4. Time-Alignment Diagnostics

The diagnostic protocol in Table S4 holds the checkpoint, task initialization, target-generation rule, noise seed, action horizon, executed prefix, geometric scales, cap, and agent budget fixed within each paired comparison. One-call diagnostics additionally replay identical observations and targets to isolate action generation. Fixed-final-index, uniform, and hard-minimum variants use, respectively, $q_h = \mathbf{1}_{h=H}$, $q_h = 1/H$, and a one-hot minimum-error assignment with earliest-index tie breaking. Fixed-temperature soft assignment uses $\beta = 4$. The annealed fixed-amplitude variant uses $\beta_t$ and envelope $0.24$; the fixed-temperature matched-envelope variant uses $\beta = 4$ and $t(1-t)$. Full soft-time guidance combines the annealed assignment and this envelope. Each variant retains the same gain and relative cap. An additional matched-correction analysis rescales per-step corrections to the full method's logged RMS, isolating assignment changes from applied magnitude.

For each generated chunk, record normalized entropy $\mathcal{H}(q)/\log H$, selected index $\arg\max_h q_h$, minimum complete-pose error, active correction RMS, and whether the arrival index lies inside the executed prefix. Temporal switches count changes in the selected index between consecutive active generation evaluations. Error and switching summaries are first averaged within each task and then across tasks; success uses the benchmark's task-level metric. The same seeds are used across variants, with five independent RSI runs and twenty evaluation seeds per task. Development tasks are used for the shared configuration; evaluation tasks provide the paired outcomes.

## C. Camera Reprojection and Visual-Geometric Feedback

### C.1. Coordinate Transforms and Projection

All returned poses use the robot-base frame $b$. Let ${}^cT_b$ map base-frame points to the optical frame of camera $c$, whose axes point right, down, and forward. For intrinsics $K_c$,

$$[p_h^c; 1] = {}^cT_b[p_h^b; 1], \qquad \widetilde{u}_h = K_c p_h^c, \quad \text{(C.1)}$$

$$u_h = (\widetilde{u}_{h,1}/\widetilde{u}_{h,3},\ \widetilde{u}_{h,2}/\widetilde{u}_{h,3})\,. \quad \text{(C.2)}$$

The renderer retains points with positive optical depth. Out-of-image segments are clipped to the image bounds; a visibility flag records their original status. RGB and depth are rectified into the same optical frame before projection. The stored intrinsics describe these rectified images, and the display-resize transform is applied consistently to both pixels and intrinsics.

**Calibration and synchronization.** Simulation adapters read intrinsics, camera transforms, and exposure timestamps from the environment API. In the Dexmal reference procedure, each camera's intrinsic calibration uses thirty checkerboard views covering the image, distance range, and board orientations. Eye-in-hand extrinsics use thirty synchronized robot/board configurations and a rigid hand–eye fit; ten disjoint configurations assess reprojection. Calibration is accepted when the held-out median reprojection error is below 2 pixels. A fixed external camera uses the corresponding board-to-base rigid registration. Calibration files store image size, distortion model, residuals, and a version identifier.

Table S4. Temporal alignment under a matched guidance envelope. LP, Dojo, and Casa denote the three simulation benchmarks. $E_{\min}$ is the terminal complete-pose energy; switches count changes in the best arrival index during one call. Mean ± SD is over five independent runs.

| Assignment | LP | Dojo | Casa | $E_{\min}$ | Switches |
|---|---|---|---|---|---|
| Fixed final step | $96.8 \pm 0.1$ | $39.1 \pm 1.7$ | $83.0 \pm 1.5$ | $1.45 \pm 0.11$ | $0.00 \pm 0.00$ |
| Uniform over steps | $95.6 \pm 0.2$ | $34.7 \pm 1.5$ | $79.8 \pm 0.9$ | $1.25 \pm 0.06$ | $0.00 \pm 0.00$ |
| Hard minimum | $98.4 \pm 0.2$ | $44.2 \pm 1.4$ | $86.1 \pm 1.1$ | $0.66 \pm 0.03$ | $1.91 \pm 0.05$ |
| Fixed temperature | $98.9 \pm 0.2$ | $45.7 \pm 4.6$ | $86.5 \pm 1.2$ | $0.68 \pm 0.06$ | $0.93 \pm 0.03$ |
| Annealed, fixed amplitude | $99.2 \pm 0.2$ | $47.8 \pm 3.3$ | $88.5 \pm 1.1$ | $0.54 \pm 0.02$ | $0.85 \pm 0.02$ |
| Fixed temperature, matched envelope | $98.9 \pm 0.3$ | $47.2 \pm 2.2$ | $87.9 \pm 1.9$ | $0.62 \pm 0.04$ | $0.96 \pm 0.03$ |
| Full soft-time | $99.4 \pm 0.2$ | $48.6 \pm 1.9$ | $89.4 \pm 0.9$ | $0.44 \pm 0.02$ | $0.73 \pm 0.09$ |

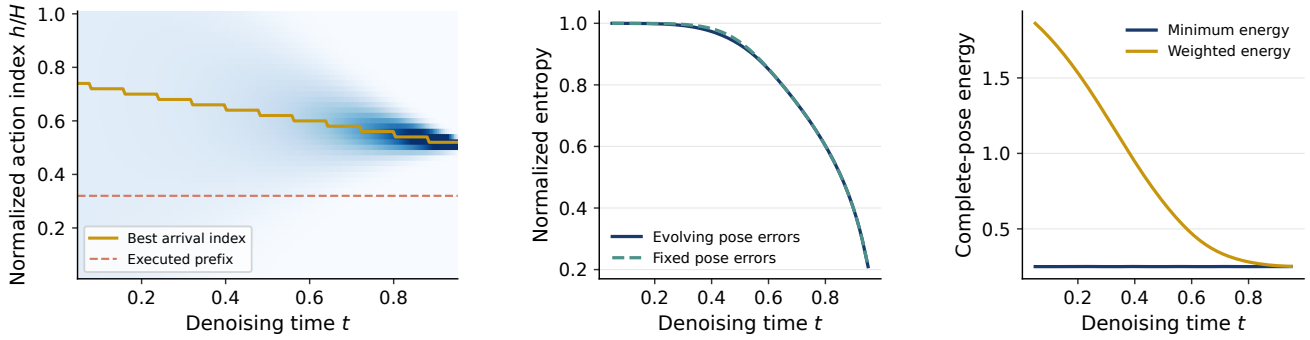



Figure S2. **Soft-time assignment over generation.** Analytical diagnostics show temporal weights, assignment entropy, selected arrival index, and minimum pose discrepancy for a fixed error profile and an evolving profile. The fixed profile isolates temperature-induced concentration; the evolving profile additionally shifts the best-matching step. Horizon and prefix markers locate the selected arrival relative to the executable approach segment; a later arrival is approached through prefix execution and renewed observation.

For a wrist camera, its transform at image exposure is computed from measured joints and the calibrated mounting transform. Robot states bracketing exposure are interpolated linearly in translation and by quaternion SLERP in rotation. The reference synchronization tolerance is 50 ms. Larger mismatch returns a stale-observation flag and triggers a fresh acquisition before geometric refinement. Both commanded poses and measured poses carry timestamps, so the agent can associate a proposed segment with the observations acquired during its execution.

**Depth and grounding.** For rectified camera-axis depth $z$,

$$p^c = zK_c^{-1}[u, v, 1]^\top, \qquad [p^b; 1] = ({}^cT_b)^{-1}[p^c; 1]. \quad \text{(C.3)}$$

If a sensor reports ray range $r$, the adapter instead uses $p^c = r\, d/\|d\|$, where $d = K_c^{-1}[u, v, 1]^\top$. The depth convention is required metadata. At a selected pixel, valid finite depths in a $5 \times 5$ neighborhood are intersected with the agent-predicted object region, and their median supplies the surface estimate. At least five valid samples are required. The median absolute deviation and valid fraction accompany the estimate; insufficient support requests another view or a newly selected point. The target generator converts this surface point into an interaction target using the approach and grasp construction in Sec. D.

## C.2. Returned Feedback and Visualization

Each accepted proposal returns its numerical command sequence and geometric visualization. The reference renderer uses one fixed external view and one wrist view when available, using a second calibrated task view when a wrist view is unavailable, preserving aspect ratio with a maximum display long side of 512 pixels. Computation uses the native calibration and resolution. The overlay shows command positions and three colored orientation axes, with red, green, and blue denoting local $x$, $y$, and $z$. A gripper-width glyph encodes the opening. Executed segments are green, unexecuted segments are blue, and the agent-specified target is an orange cross. The selected arrival index and the execution-prefix boundary receive distinct circle and square markers.

At most twelve evenly spaced pose glyphs are drawn, supplemented by the selected arrival and prefix-boundary indices. The full numerical chunk remains available. Depth

Table S5. **Action-informed feedback.** Every response includes proposal and observation identifiers. Numerical arrays use the adapter's declared units and frame.

| Field | Representation / source |
|---|---|
| Generated chunk | Full $H \times D$ command array, mask, seed, and $H$ decoded complete poses |
| Executed prefix | Index range, timestamps, command log, measured states, and per-command status |
| Continuation | Unexecuted indices and original command poses; state version for rebasing |
| Projection | Up to two rectified views; sampled pose glyphs, target, depth visibility, and calibration ID |
| History | Last three available observation/state records with exposure and execution timestamps |
| Proprioception | Joint positions, measured end-effector poses, gripper widths, and controller status |
| Compatibility | $C_t$, pose residual, relative correction demand, checkpoint index, and thresholds |
| Error / rejection | Structured status, failing operation or exceeded tolerance, and executed-action count |

also supports visibility: a projected command point more than 2 cm behind the observed surface is drawn with a dashed segment and an occlusion flag. This annotation describes the visible scene geometry associated with the command. Action execution and observed object motion are supplied by the timestamped history. The agent can inspect approach direction, rotation, gripper timing, and the relation between the executed prefix and remaining proposal.

After execution changes the state, a stored continuation remains associated with its original proposal. Reusing it creates a newly decoded and rebased candidate from current proprioception, with new projections and a new version. The tool requires explicit execution of that new candidate. This correspondence keeps visual feedback, commanded motion, and actual history aligned during local refinement.

### C.3. Calibration and Feedback Robustness

The robustness protocol perturbs one factor at a time while preserving task initialization, target-generation code, action noise, and agent budget. Extrinsic translations have magnitudes $\{0, 5, 10, 20\}$ mm with uniformly sampled directions; extrinsic rotations use angles $\{0, 1, 3, 5\}^\circ$ around uniformly sampled axes. Depth noise is independent zero-mean Gaussian with standard deviation $\{0, 5, 10, 20\}$ mm, applied to valid depth pixels before the neighborhood estimator. The same perturbed calibration is used consistently for target grounding and proposal overlays, while the environment and robot geometry retain their original calibration.

Occlusion masks cover fractions $\{0, 0.10, 0.25, 0.40\}$ of the grounded object's image region, with positions sampled independently of the target pixel. Missing-depth handling follows the same valid-support rule. Feedback delay uses $\{0, 100, 250, 500\}$ ms and retains the delayed frame's original timestamp, activating the synchronization rule when necessary. This evaluates the complete feedback pipeline, including reacquisition and its cost. A separate bypass condition disables reacquisition at the same delay levels to isolate the contribution of timestamp handling.

Table S6 organizes task success, target-localization error, and recovery cost under these controlled conditions. Localization error is the Euclidean displacement of the estimated interaction point from its unperturbed reference, measured before action generation. Recovery cost counts all additional agent calls within the episode, including requests for fresh observations. Results are aggregated over matched task seeds, with per-task paired differences and task-level bootstrap confidence intervals. These diagnostics connect geometric perturbations to both execution outcomes and the amount of corrective interaction.

## D. Agent Interface, Target Generation, and Local Refinement

### D.1. Observation Access and Geometric Grounding

The reference tool interface presents a timestamped observation packet containing the instruction, RGB views, registered depth, camera calibration, measured joint and end-effector states, gripper state, and a monotonically increasing state identifier. Table S7 defines the observation contract. Every matched agent configuration receives the same packet. Object poses, object identities from simulator internals, reward terms, and contact annotations are recorded by the evaluator. A separate reference-target condition explicitly enables demonstration or simulator target poses. The normal agent constructs targets from its observation packet. Regions and keypoints are agent predictions; evaluator object masks and annotations remain in the scoring record.

For each intended interaction, the agent chooses a visible object region and a contact pixel, and invokes calibrated back-projection. The tool uses the median valid depth in a $5 \times 5$ pixel neighborhood within that region, rejecting a neighborhood with fewer than five valid samples. The position is expressed in the declared robot-base frame. An approach direction and a task-relevant tangent define an orthonormal target frame by Gram–Schmidt orthogonalization. A tangent within $10^{-3}$ of parallel to the approach direction triggers selection of the least-aligned base axis as the auxiliary vector. The interface converts the resulting rotation to the checkpoint's orientation representation. The agent chooses the gripper command and constrains the active arm through component masks. An initial approach offset of 5 cm along the negative approach direction provides a pre-

Table S6. Visual geometry stress test on RoboDojo. Each factor uses the two listed severities with equal weight; all other settings remain fixed. Error is the target-grounding distance to the corresponding simulator reference. Mean ± SD is over five independent runs.

| **Perturbation** | **Severity pair** | **SR (%)** | **Error (mm)** | **Calls** |
|---|---|---|---|---|
| None | 0 | $48.6 \pm 1.9$ | $10.2 \pm 0.4$ | $7.8 \pm 0.7$ |
| Extrinsic translation | 5 / 20 mm | $42.0 \pm 2.3$ | $19.9 \pm 0.9$ | $9.1 \pm 0.5$ |
| Extrinsic rotation | 1 / 5 deg | $41.2 \pm 3.8$ | $33.5 \pm 2.1$ | $9.8 \pm 0.5$ |
| Depth noise | 5 / 20 mm | $42.1 \pm 3.8$ | $24.4 \pm 1.2$ | $8.9 \pm 0.4$ |
| Missing depth | 10 / 40% | $39.5 \pm 1.9$ | $28.5 \pm 1.6$ | $10.5 \pm 0.9$ |
| Observation delay | 100 / 500 ms | $38.5 \pm 2.0$ | $36.7 \pm 0.9$ | $11.6 \pm 0.6$ |

Table S7. Observation contract for the reference implementation. The adapter resolves camera names against the environment configuration and records the selected names and calibration in the run manifest.

| Platform | Visual observations | Robot observations |
|---|---|---|
| LIBERO-Plus | Fixed view, wrist view, depth | EEF pose, gripper |
| RoboDojo | Two calibrated task views, depth | Both arms, both grippers |
| RoboCasa365 | Fixed view, wrist view, depth | EEF pose, gripper |
| Dexmal | Fixed view, wrist view, depth | Joint/EEF state, gripper |

contact target. The subsequent contact target is issued in a new call.

When several contacts are plausible, the agent ranks at most three candidates using visible accessibility, consistency with the task phase, and prior failed attempts. It requests one candidate at a time. Each target includes its source observation ID, selected pixel or region, coordinate frame, and a brief intended effect. A target grounded in an earlier state is rechecked after an execution event or a new object observation. In sequential bimanual manipulation, the agent issues ordered calls with an explicit active-arm field and a hold target for the supporting arm.

## D.2. Tool Schemas and Control Permissions

The tool protocol separates proposal generation from execution. All mutations carry a unique request ID, observation ID, and expected robot-state version. Retrying a request with the same ID returns the cached result. The executor validates the state version before consuming a proposal. An outdated proposal is regenerated or explicitly rebased using a new observation. Execution occurs serially, and a stop request cancels the remaining queued commands. Each JSON request has a schema version, string request ID, integer state version, tool name, and typed argument object. Target arguments contain a three-vector position in metres, a rotation in the adapter representation, one normalized gripper value per active arm (0 closed, 1 open), an explicit frame ID, Boolean component masks, and a source observation ID. The parser requires finite values, a valid rotation, a nonempty active mask, and a horizon matching the registered adapter. Returned status is one of accepted, rejected, stale, execution-failed, or completed.

**Local edits.** The edit tool operates on an unexecuted segment of an accepted chunk. The agent specifies its first and last indices and a translation $\delta p$ and rotation vector $\delta\omega$. The reference limits are $\|\delta p\|_2 \leq 0.02\,\mathrm{m}$ and $\|\delta\omega\|_2 \leq 10^\circ$. For a segment with at least three samples, let $s_h = (h - h_a)/(h_b - h_a)$ and define

$$b(s) = 16s^2(1-s)^2, \tag{D.1}$$

$$p'_h = p_h + b(s_h)\delta p, \tag{D.2}$$

$$R'_h = \mathrm{Exp}(b(s_h)\delta\omega)\, R_h. \tag{D.3}$$

The correction has zero value and derivative at both boundaries, so it joins the original motion without introducing a position or velocity jump in the continuous correction profile. Because $0 \leq b(s) \leq 1$, the maximum offset respects the stated edit limits. Gripper-transition edits move an existing transition by at most two action indices and retain at least two commands in each affected state. Inactive-arm commands are copied from the original chunk.

The native-action adapter converts the edited poses back to executable commands. A joint-space adapter uses sequential damped least-squares inverse kinematics, initialized from each original joint command, with damping $10^{-3}$, at most 30 iterations per pose, and a $0.05$ rad maximum joint update. Every solution must satisfy joint limits and residuals below 2 mm and $2^\circ$. The tool returns an edit failure if this solve fails. Delta Cartesian adapters apply their inverse composition convention. Controller bounds and command-rate limits are checked after conversion.

The agent selects regeneration when the object, interaction phase, or approach strategy changes. It selects local editing when the proposed maneuver is useful and the discrepancy lies within the edit envelope. If two executed prefixes reduce the measured target error by less than 20%, the agent refreshes the observation and revises the target. The

Table S8. Reference agent-facing tool contract. AV denotes Agent + VLA, AI denotes Agent + InteractVLA, and AO denotes Agent Only. Trace access is varied explicitly in the feedback ablation. All methods have the same direct-control and stop permissions.

| Tool | Required payload | Returned fields | Access |
|---|---|---|---|
| `observe` | Camera IDs, history length | State ID, RGB-D, calibration, proprioception | AV, AI, AO |
| `backproject` | View/state ID, pixel/region, approach | Base-frame contact, valid-depth count | AV, AI, AO |
| `propose` | Instruction, arm mask, $H$, observation ID | Chunk ID, status, model/seed IDs | AV, AI |
| `guide` | Propose fields plus $G$, component masks | Chunk ID, $r_t, d_t, C_t$, status, trace | AI |
| `inspect` | Chunk ID, view IDs, step indices | Command poses, projections, execution flags | AI / ablation |
| `edit` | Chunk ID, index range, pose offsets | Revised chunk ID, edit report, validity | Trace-enabled |
| `execute` | Chunk ID, prefix length, state version | Executed indices, elapsed time, new state ID | AV, AI |
| `control` | Cartesian waypoint, gripper, duration | Executed command, status, new state ID | AV, AI, AO |
| `stop` | Request ID | Queue status, final observed state | AV, AI, AO |
| `memory` | Read/write mode, task key, version | Retrieved records or immutable new version | AV, AI, AO |

inspect tool overlays 12 uniformly sampled poses, the selected arrival index, and the prefix boundary; all native commands remain available numerically. Direct-control commands remain available through the shared `control` tool. The ledger records the number and duration of these commands separately from policy-generated execution, allowing matched comparisons of control permissions and action feedback.

## D.3. Prompts and API Configurations

The following literal templates define the reference agent. Braced uppercase fields are populated from typed tool outputs. Task-specific content enters through the instruction, observations, adapter metadata, and retrieved experience.

**System prompt.**

> You control a robot through the listed tools. Solve the current task using the available observations and robot state. Use the stated coordinate frames and units. Before acting, identify the next interaction and its intended effect. Prefer a VLA motion proposal when its manipulation skill fits the interaction. Use observed outcomes to decide whether to continue, revise the target, or edit a local motion segment. Keep supporting arms consistent with the task. Submit one structured tool request at a time. Stop when the task is complete or the execution budget is exhausted. Store concise evidence about successful and unsuccessful decisions.

**Task and target-generation prompt.**

> Instruction: {TASK}. Current packet: {OBSERVATION}. Robot conventions: {ADAPTER}. Available experience: {MEMORY}. Select the next interaction. Return a tool request with instruction, observation ID, active arm, horizon, and intended effect. For a guided call, add target position in metres, rotation in the adapter representation, gripper command, component masks, and the pixel/region used for grounding. Resolve target coordinates with the back-projection tool. Choose a reachable pre-contact pose before a contact pose when an approach is needed. Use a new observation to resolve an ambiguous contact candidate. Output a JSON object conforming to the chosen tool schema; for example, a guide request includes instruction, arm, horizon, target, masks, frame, and observation ID. Obtain the numerical target from the current grounded observation.

**Motion-diagnosis and recovery prompt.**

> Previous intent: {INTENT}. Proposal and execution record: {TRACE}. Current observation: {OBSERVATION}. Compare the proposed approach, orientation, and gripper timing with the observed outcome. Identify the affected action interval. Return one decision: execute the remaining permitted prefix, regenerate using a revised target, apply a bounded local edit, request an observation, issue a direct-control command, or stop. Include the visual evidence and intended correction. For rejection, use the reported pose residual and correction demand to revise the candidate or split the interaction into shorter stages.

**RSI revision and validation prompts.**

> Review {EPISODE_LEDGER} and strategy {VERSION}. Propose one explicit revision to target selection, stage order, recovery, or memory retrieval. Return the triggering condition, revised behavior, supporting attempt IDs, and expected effect. Submit the candidate

to the fixed validation procedure. Compare candidate and parent results using the registered acceptance rule. Commit an accepted candidate as a new immutable version. Otherwise retain the parent and record the failed hypothesis. During final evaluation, retrieve experience from the selected strategy version and record outcomes separately.

**Context management.** Each call receives the latest frame from at most two views, resized to a maximum image dimension of 512 pixels with the projection calibration transformed accordingly, plus the latest three execution-boundary snapshots. Numeric targets, poses, and execution flags accompany the images. The reference text budget is 24,000 tokens. The latest eight tool events are retained verbatim. Older events are condensed into one record per interaction, preserving target coordinates, decisions, outcomes, and event IDs. Retrieval returns the four most relevant records using task-key match followed by text similarity and deterministic ID tie-breaking. Summarization and retrieval calls enter the API and timing ledger.

**API configuration.** The experiment configuration names the agent family (Sol, Astra, or Claude Fable 5.2), provider, exact deployment/model ID, API version, reasoning setting, sampling parameters, and execution dates. The runner obtains the resolved model ID from each response and stores it with the request configuration. The reference final-output limit is 4,096 tokens. Temperature is 0.2 and reasoning effort is high for deployments exposing these controls. For deployments with a fixed setting, the adapter records the effective provider setting. Structured tool arguments are validated against the same schema for every agent. Malformed outputs receive one schema-correction retry. Transport failures receive at most two retries with 1 and 4 second delays. Request IDs prevent duplicate physical execution. Prompt caching uses identical static prefixes, with cached and uncached input tokens logged separately.

**Benchmark adaptation.** The language templates and decision tools are shared. Adapter changes specify native action dimensions, camera names, coordinate transforms, arm count, valid command bounds, and environment termination predicates. Per-task instructions come from the benchmark; the reference configuration introduces no manually written task-specific strategy. An operator reset or intervention terminates the current evaluation episode and is retained in its outcome record. The episode and RSI budgets follow Section F. A new run records the hashes of its templates, adapter, and tool schema, as well as all operator resets and interventions.

# E. Target Compatibility and Early Rejection

## E.1. Score and Decision Rule

A compatibility checkpoint assesses the current clean-action estimate before the next guided update. With $E_h$ from the complete-pose objective and $M$ selecting valid normalized action coordinates, define

$$r_t = \min_h E_h, \tag{E.1}$$

$$d_t = \frac{\lambda\,\mathrm{RMS}_M(g_t)}{\mathrm{RMS}_M(v_\theta) + \epsilon_v}, \tag{E.2}$$

$$C_t = \exp\left[-\max\left(\frac{r_t}{r_0}, \frac{d_t}{d_0}\right)\right]. \tag{E.3}$$

The RMS uses the same active-valid mask $M$ as Eq. A.16, excluding padding and inactive arms. The score uses the requested gradient before the correction cap. Thus $d_t$ measures the instantaneous relative correction requested at the current state. Equivalently, for native increment $a = \Delta t\, v_\theta$ and requested guided increment $b = \Delta t(v_\theta - \lambda M \odot g_t)$,

$$d_t = \frac{\mathrm{RMS}_M(b-a)}{\mathrm{RMS}_M(a) + \Delta t\,\epsilon_v}. \tag{E.4}$$

The reference defaults are $r_0 = 1$, $d_0 = 3$, and $\epsilon_v = 10^{-3}$ in normalized action-velocity units. Selection uses 50 development states excluded from the gate evaluation. The threshold and both scales are frozen before evaluation.

**Decision-boundary proposition.** For finite nonnegative $r_t, d_t$ and positive $r_0, d_0$, $C_t \in (0, 1]$ is nonincreasing in either discrepancy. For any cutoff $\kappa \in (0, 1)$,

$$C_t \geq \kappa \iff \begin{cases} r_t \leq -r_0 \log \kappa, \\ d_t \leq -d_0 \log \kappa. \end{cases} \tag{E.5}$$

Indeed, taking logarithms gives $\max(r_t/r_0, d_t/d_0) \leq -\log\kappa$, which is exactly the conjunction of these two inequalities. At $\kappa = e^{-1}$, continuation therefore requires $r_t \leq r_0$ and $d_t \leq d_0$. Equality continues generation. A small correction demand cannot compensate for a large complete-pose residual in this maximum-based rule. The score expresses compatibility of the current proposal with the target under the declared tolerances. Its relationship to physical success is evaluated by the paired rollout procedure below.

**Checking schedule and return contract.** For the ten-step Euler schedule, the reference active window contains $t = 0.5, 0.6, 0.7$. The first guided update occurs at $t = 0.5$. The first checkpoint evaluates the resulting prediction at $t = 0.6$, and a second checkpoint precedes the update at $t = 0.7$. Each checkpoint computes the current input gradient for its demand score. A call rejected at $t = 0.6$ therefore

has seven expert forward evaluations and two backward evaluations, at $t = 0.5$ and $t = 0.6$. Rejection ends sampling before the pending update and before any command from that call executes. Nonfinite predictions or gradients produce a numerical-failure status and the same no-execution return path.

The return record contains `status`, `reason`, state and chunk IDs, generation step, minimum-error action index, $r_t$, $d_t$, $C_t$, the requested target, partial proposal, and compute/API counters. Reasons distinguish pose mismatch, excessive correction demand, both conditions, and numerical failure. Rejected proposals support inspection. The executor accepts only chunk IDs with an accepted status and a matching state version.

After three consecutive rejected calls, the agent obtains fresh observations and revises the interaction stage. The consecutive-rejection counter resets after an accepted generation, while observation requests leave it intact. Six consecutive rejections terminate the episode with a recorded rejection-budget outcome. The overall episode budget in Section F also remains active. Every rejected generation and subsequent recovery call contributes to reported cost.

### E.2. Ablation and Threshold Sensitivity

The gate study samples 100 RoboDojo states, with 20 from each capability category. Each state has an observation-grounded target and four perturbations consisting of two translations and two rotations. The translations have magnitudes 5 and 15 cm along seeded random unit directions. The rotations have magnitudes $15^\circ$ and $45^\circ$ about seeded unit axes. Two matched Gaussian initializations give 1,000 state–target–noise calls. Task IDs and reset seeds are shared across gate conditions and separated from the development set.

For each call, a simulator copy continues the ungated sampler and executes the complete $H$-step proposal under the same state and noise seed. The subgoal predicate is fixed before comparison and evaluated over this full-horizon diagnostic rollout. Deployment retains the configured executed-prefix length $H_{\text{exec}}$. Let $Y_i = 1$ if that continuation satisfies the subgoal and let $R_i = 1$ if the gate rejects the corresponding proposal. Rejection rate is $\sum_i R_i/n$. False rejection is $\sum_i R_i Y_i / \sum_i Y_i$, and accepted-call failure is $\sum_i (1 - R_i)(1 - Y_i) / \sum_i (1 - R_i)$. These definitions identify both the accepted-call risk and the cost of rejecting viable continuations. An empty denominator is reported as undefined.

The four gate conditions are always continue, pose discrepancy alone, correction demand alone, and the combined score. The component-only rules retain the corresponding normalized tolerance from the combined rule; their held-out coverage is reported alongside their error rates. A second evaluation reruns the full agent on the complete RoboDojo task manifest with 20 reset seeds per case and the same episode budgets, measuring task success and end-to-end time including rejected calls and replanning. Proposal-level bootstrap intervals resample the 100 initial states, retaining all associated target variants and noise seeds. Full-agent intervals instead resample task cases, retaining all 20 reset seeds and paired methods within each case.

The threshold sweep fixes $r_0, d_0, \epsilon_v$ and uses $\kappa \in \{0, 0.1, e^{-2}, 0.2, e^{-1}, 0.5, 0.7\}$. The zero cutoff disables rejection. Every positive cutoff admits normalized discrepancies up to $-\log \kappa$; in particular, $e^{-2}$ and $e^{-1}$ admit discrepancies up to 2 and 1. On fixed ungated traces, increasing the cutoff produces a nested set of rejected calls, so rejection is nondecreasing. Full-agent success and cost are evaluated by rerunning each cutoff because target revision changes subsequent states and proposals. This separates the proposal-level threshold response from the effect of recovery on the final task outcome.

## F. Benchmark Protocols, Baselines, and Statistical Reporting

### F.1. Benchmark-Specific Settings

The reference evaluation separates online tool use, practice with recursive self-improvement (RSI), and evaluation of the resulting fixed strategy. Every simulation condition uses five independent runs: online experiments vary their random-seed namespaces, and RSI experiments independently acquire five strategy snapshots. Table S11 specifies the controlled studies. Each run begins by exporting a task manifest containing the benchmark release, repository commit, task and variant IDs, category membership, object and scene assets, initialization seeds, control frequency, horizon, and success-predicate version. The evaluator hashes the manifest, checkpoint, adapter, and configuration before launching any method. This makes task coverage an executable specification shared by every compared system.

**LIBERO-Plus.** The full-manifest evaluation retains background, lighting, language, sensor noise, layout, camera, and robot-initialization perturbations, together with every severity and variant declared by the selected release [3]. The manifest records the parent task of each perturbation, allowing both per-category and per-parent-task summaries. Each trial starts from a fresh simulator reset; policy history and episode-local agent state are cleared. The simulator's task-success predicate terminates successful execution, and its control timestep is retained. The common-protocol comparison additionally uses the exact 120-variant ID list associated with the HarnessVLA and RV-ICL comparison [1, 10]. The list and its hash identify that experiment, while the full-manifest results retain their separate coverage label.

Table S9. Compatibility-gate components. Rejection, false rejection, and accepted-call failure use 1,000 paired call proposals; success and time use the separate full-agent rollout cohort. False rejection is conditional on viable ungated continuation. Mean ± SD is over five independent runs.

| Gate | SR (%) | Reject (%) | False reject (%) | Accepted failure (%) | Time (s) |
|---|---|---|---|---|---|
| No rejection | $47.2 \pm 1.4$ | 0.0 | 0.0 | 36.3 | $200 \pm 12$ |
| Pose discrepancy only | $47.1 \pm 1.4$ | 30.3 | 5.3 | 13.5 | $208 \pm 18$ |
| Correction demand only | $46.6 \pm 2.0$ | 24.4 | 2.5 | 17.9 | $200 \pm 12$ |
| Combined score | $48.6 \pm 1.9$ | 38.6 | 7.8 | 4.4 | $201 \pm 6$ |

Table S10. Confidence threshold sweep with $r_0 = 1$ and $d_0 = 3$. Coverage is the fraction of accepted proposals. Full-agent success includes replanning after rejection; accepted-call failure uses fixed ungated counterfactuals. Mean ± SD is over five independent runs.

| $\kappa$ | Coverage (%) | SR (%) | False reject (%) | Accepted failure (%) | Time (s) |
|---|---|---|---|---|---|
| 0 | 100.0 | $47.2 \pm 1.4$ | 0.0 | 36.3 | $200 \pm 12$ |
| 0.10 | 87.2 | $46.5 \pm 2.3$ | 0.3 | 27.2 | $201 \pm 11$ |
| $e^{-2}$ | 83.9 | $47.6 \pm 3.9$ | 0.3 | 24.3 | $207 \pm 9$ |
| 0.20 | 76.8 | $47.5 \pm 1.4$ | 1.1 | 18.0 | $196 \pm 12$ |
| $e^{-1}$ | 61.4 | $48.6 \pm 1.9$ | 7.8 | 4.4 | $201 \pm 6$ |
| 0.50 | 51.3 | $47.0 \pm 2.3$ | 20.7 | 1.6 | $220 \pm 13$ |
| 0.70 | 20.9 | $42.9 \pm 3.3$ | 67.2 | 0.0 | $259 \pm 14$ |

**RoboDojo.** The manifest assigns each official task to Generalization, Precision, Long-Horizon, Memory, or Open, and includes the scene-layout and object-randomization arguments used by its reset function [2]. All methods receive the same camera streams, robot state, native controller interface, timeout, and task instruction. Agent + VLA, Agent + InteractVLA, and Agent Only share observation, direct-motion, and gripper tools; the policy interface is the experimental factor. The added cross-model study executes exactly 20 seeded trials for every official task. Its repetition count is recorded alongside the task coverage, separately from any archived main-paper or official-submission protocol.

**RoboCasa365.** The multitask manifest contains 18 Atomic-Seen, 16 Composite-Seen, and 16 Composite-Unseen tasks. The benchmark's seen/unseen labels concern task membership in Human300 pretraining; Composite-Unseen holds out composite task definitions [8]. The reference RSI experiment adds an independent practice/evaluation partition over kitchen, object, layout, and seed tuples. An additional composition-transfer condition holds out complete task compositions from RSI practice as well. Training-data access, kitchen lists, and memory versions are logged separately. For comparison with HexaShell 1.5, the release record identifies its disclosed pretrain50 kitchen set and task-memory configuration [4]; a matched rerun uses the intersection of those kitchen IDs with the selected benchmark release. The agent observes policy inputs, and the evaluator releases the binary success outcome only after an episode ends.

**Aggregation.** For task $i$, trial $j$, and independent practice run $r$, let $s_{rij} \in \{0, 1\}$ denote success. The controlled studies use equal task weights,

$$\widehat{\mathrm{SR}} = \frac{100}{R} \sum_{r=1}^{R} \sum_{i=1}^{T} \omega_i \frac{1}{n_i} \sum_{j=1}^{n_i} s_{rij}, \quad \omega_i = \frac{1}{T}. \tag{F.1}$$

Category summaries renormalize these weights within a category. RoboCasa365 therefore weights its three split means by $18/50$, $16/50$, and $16/50$. LIBERO-Plus treats a task–perturbation variant as the evaluation unit and also exports a parent-task summary. An official-score export invokes the selected release's aggregation routine and stores its resulting task weights. Every score record identifies its aggregation and denominator. The historical cost cohorts of 505, 450, and 9,436 recorded cases define cost coverage; task coverage comes from the manifests above.

**Shared reference configurations.** Supplementary expansions of a main-paper condition preserve its reported mean and model configuration. The complete-RSI reference profile uses Sol with MR-0 on LIBERO-Plus (99.4%), Astra with MR-0 on RoboDojo (48.62%, displayed as 48.6% at

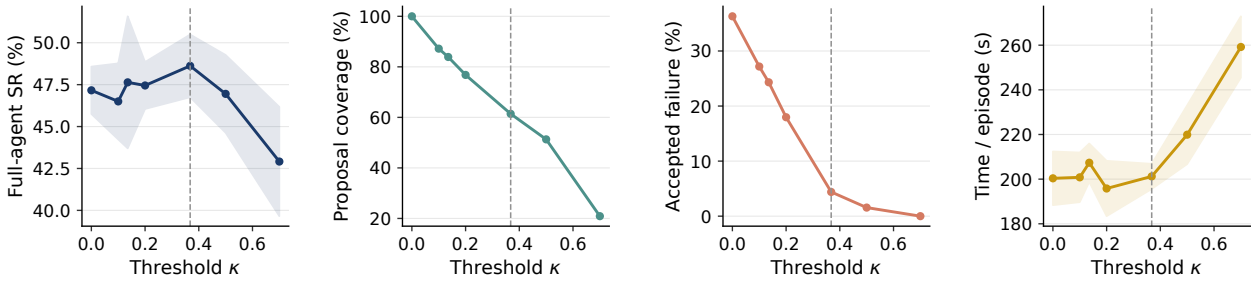



Figure S3. **Compatibility threshold controls the acceptance–recovery tradeoff.** Coverage and accepted-call failure are evaluated on matched proposal traces. Task success and time are evaluated by rerunning the complete agent at each cutoff, including target revision after rejection. The default cutoff is $e^{-1}$.

Table S11. Reference evaluation settings. Simulation conditions have five independent runs. Ordinary conditions use 20 evaluation seeds per task or variant per run; cross-model conditions allocate four disjoint seeds to each run, totaling exactly 20 trials per task. RSI has ten practice attempts per case in each run. Episode limits count executed controller steps; API limits apply within each attempt or evaluation episode.

| Condition | Coverage | Trials | Agent | Backbone | Practice | Episode / API limit |
|---|---|---|---|---|---|---|
| LIBERO-Plus tool use | Complete variant manifest | 20/run | Sol | MR-0 | 0 | 1,000 steps / 64 |
| RoboDojo tool use | Complete task manifest | 20/run | Sol | MR-0 | 0 | 2,000 steps / 128 |
| LIBERO-Plus RSI | Complete variant manifest | 20/run | Sol | MR-0 | $5 \times 10$ | 1,000 steps / 64 |
| RoboDojo RSI | Complete task manifest | 20/run | Astra | MR-0 | $5 \times 10$ | 2,000 steps / 128 |
| RoboCasa365 RSI | 50-task multitask manifest | 20/run | Astra | MR-0 | $5 \times 10$ | 2,000 steps / 128 |
| RoboDojo cross-agent | All official tasks | 20 total | Sol / Astra / Fable | MR-0 | $5 \times 10$ | 2,000 steps / 128 |
| RoboDojo cross-backbone | All official tasks | 20 total | Fable | MR-0 / $\pi_{0.5}$ | $5 \times 10$ | 2,000 steps / 128 |
| Dexmal | Four manipulation tasks | 30 | Astra | MR-0 | 0 | 600 seconds / 128 |

one-decimal precision), and Astra with MR-0 on RoboCasa365 (89.4%). Repeated reference rows share these main-paper summaries; additional controlled comparisons retain their own protocol and result records. The provenance index records the numerical source of each mean and accompanying variability summary.

## F.2. Baseline Provenance and Matched Comparisons

Table S12 separates the source of an external score from the specification of a controlled implementation. An in-house Agent + VLA row is a language-only policy interface under the shared agent harness. Its identity includes the prompt, tools, checkpoint, memory, and budget; named external systems retain their own identity and source.

The common-protocol study fixes Sol, the MR-0 checkpoint, the 120-variant ID list, 20 paired evaluation seeds per variant, and the same ten-attempt practice allowance. A method adapter preserves its memory and decision mechanism while using the shared observation and robot-control schema. Each adapter record lists changed prompts, unavailable native operations, input transformations, and memory retrieval. Hyperparameter selection has a maximum of 20 configurations evaluated on the same five development instances per configuration; the selected configuration is fixed before evaluation. Success signals, reference poses, simulator state, and human interventions have separate access flags, allowing the comparison to be reconstructed directly from the ledger.

## F.3. Randomness, Uncertainty, and Failures

Environment, action-noise, agent-sampling, and run seeds occupy separate namespaces. A seed is the first 32 bits of SHA-256 applied to the release hash, task ID, split, run index, trial index, and seed namespace. The same environment and action-noise seeds pair interface variants within each run; provider request IDs and stochastic decoding settings are saved for each agent call. Five independent online runs measure seed variation, while five independently initialized RSI chains capture end-to-end variation from practice and evaluation randomness.

Controlled simulation tables report success as **mean ± sample standard deviation across five independent runs**. Let $S_r$ be the task-weighted success percentage from run $r$. The reported statistics are

$$\bar{S} = \frac{1}{5}\sum_{r=1}^{5} S_r, \qquad \mathrm{SD}_{\mathrm{run}}(S) = \sqrt{\frac{1}{4}\sum_{r=1}^{5}(S_r - \bar{S})^2}. \tag{F.2}$$

The SD is in percentage points. A paired gain is computed within each run before taking its mean and SD. Other simulation metrics first aggregate within each run, then use the

Table S12. Baseline provenance and comparison design. Published entries retain their cited model and protocol. Controlled implementations use the fixed task manifest, agent observations, generic tools, and interaction budget specified in this supplement.

| Method | Source | Record type | Controlled comparison factor |
|---|---|---|---|
| Standalone MR-0 | Selected checkpoint | In-house control | Policy execution; no agent |
| Agent + VLA | Shared agent harness | In-house control | Language-only policy call |
| Agent Only | Shared agent harness | In-house control | Direct robot control |
| HarnessVLA | Paper [10] | Published score / reference adapter | Memory-guided policy tool use |
| RV-ICL | Paper [1] | Published score / reference adapter | Video experience and policy tool use |
| PhysicalRSI | Release [7] | Published score | Cited agent and policy-skill configuration |
| WorldScape-R$^2$SI | Release [6] | Published score | Cited agent and world-action-model system |
| EmbodiedEvo | Leaderboard [9] | Published score | Cited embodied-agent configuration |
| HexaShell variants | Submission [4] | Published score | Cited kitchen, task-memory, and agent setup |
| VLS | Version 1, Sec. IV [5] | Reference adapter | Reward-based sampling and stage control |

Table S13. Common-protocol LIBERO-Plus comparison on a 120-variant overlap manifest, with 20 paired seeds per variant. Tokens and steps are per evaluation episode. The agent and MR-0 checkpoint are fixed across adapters. Mean ± SD is over five independent runs.

| **Method** | **SR (%)** | **95% CI** | **Tokens (M)** | **Steps** |
|---|---|---|---|---|
| Language-only | $92.3 \pm 0.8$ | [91.4, 93.2] | $0.361 \pm 0.022$ | $272 \pm 6$ |
| HarnessVLA adapter | $94.7 \pm 0.8$ | [93.8, 95.5] | $0.272 \pm 0.015$ | $253 \pm 12$ |
| RV-ICL adapter | $96.3 \pm 0.3$ | [95.8, 96.7] | $0.288 \pm 0.026$ | $246 \pm 16$ |
| InteractVLA | $98.4 \pm 0.2$ | [98.0, 98.7] | $0.207 \pm 0.014$ | $230 \pm 17$ |

same reporting rule. Per-task success counts and all five run-level values accompany each table.

Additional 95% intervals use 10,000 paired bootstrap replicates. Each replicate samples run IDs and task IDs as crossed clusters, with task resampling stratified by category; the selected task list is shared across sampled runs. Paired trial IDs are then resampled within each run–task block, and all methods share the sampled indices. Task weights follow Eq. (F.1). A task-only bootstrap with the strategy fixed measures instance uncertainty, while run dispersion measures strategy or seed variation. Confidence intervals accompany the across-run SD as a separate uncertainty summary. Prespecified primary contrasts are the interface gain, the interface–RSI interaction, and cross-model paired gains; any family of hypothesis tests uses Holm correction at level 0.05.

**Failures and exclusions.** Timeout, invalid action, exhausted rejection or API budget, and a safety stop during execution are task failures. All consumed steps, tokens, and time remain in the ledger. Provider transport failures receive at most two transport retries with the original request and state; retries consume cost and do not reset the episode. A reset failure or simulator initialization crash before the first observation is recorded as an infrastructure-invalid trial and rerun with the same seed. A crash after execution begins remains an interrupted failure in the primary operational score, with its infrastructure cause recorded separately. The report exports every exclusion, original seed, retry, and outcome. Hardware evaluation logs operator resets between trials and counts intervention during a trial as failure.

# G. Recursive Self-Improvement and Generalization

## G.1. What Is Updated

RSI updates a versioned task-solving strategy: target-selection rules, stage ordering, recovery conditions, reusable action-edit templates, successful and failed examples, and retrieval metadata. Each record contains a task-semantic key, scene descriptors, triggering observation, target and chunk IDs, outcome, consumed budget, diagnosis, revision, and parent version. Policy parameters and the command-to-pose adapter are fixed throughout these studies. Online recovery modifies episode-local decisions; accepted RSI revisions become persistent state for later practice attempts.

**Update and validation loop.** A practice case permits ten attempts including the initial attempt, with a limit of 64 agent requests per attempt for LIBERO-Plus and 128 per attempt for the other benchmarks. After a completed attempt, the agent proposes at most one strategy revision using its observation and action history and the terminal success signal. A separate development evaluator compares the candidate and its parent on the same five seeded instances. The acceptance rule first maximizes completed tasks, then minimizes agent

calls, then minimizes end-to-end time; a strict improvement is required. A preregistered protected-skill suite additionally requires each skill's development success count to remain at least its parent value. The suite uses its own fixed five seeds per skill. The parent remains active during validation, and an accepted candidate receives an immutable version and checksum. Failed candidates retain their diagnosis and validation trace, while execution rolls back to the parent. All validation rollouts and revision calls enter the practice-cost ledger. Final selection follows the same rule after the practice budget, and the chosen strategy is fixed for evaluation.

**Retrieval and scope.** Retrieval first filters by task semantics and compatible robot/tool schema, then ranks records by scene-description similarity with stable record ID as a tie-breaker. At most four records and 8,000 text tokens enter the shared 24,000-token context budget. The eight most recent tool records are retained verbatim and earlier interactions are summarized. A retrieved example includes its original outcome and conditions of applicability. The shared policy library is read-only; task-specific memory is copied at the start of each independent practice run. Evaluation reads the selected snapshot, keeps episode-local observations, and writes outcomes to a separate analysis log. Human-authored prompts and schemas are fixed before practice; revision selection follows the evaluator's rule.

## G.2. Practice–Evaluation Separation

The split generator assigns canonical scene-instance keys by a salted deterministic hash, and writes the full resulting lists before any practice. Seeds are drawn only after asset and layout partitioning, so an instance split separates geometry as well as pseudorandom state. Three transfer conditions distinguish new scenes for practiced tasks, new object instances within familiar task semantics, and new task compositions. In composition transfer, every instance of a held-out semantic composition is assigned to evaluation. The benchmark's pretraining-seen label and the RSI-practice-seen label are stored independently, especially for RoboCasa365 Composite-Unseen. An overlap audit compares canonical asset IDs, scene graphs, demonstration IDs, and composition keys across the four manifests in Table S14.

## G.3. Separating Interface Benefits from RSI Benefits

The factorial study fixes Astra, MR-0, RoboDojo task instances, generic tools, and evaluation budget. Rows $A$ and $C$ use the initial strategy; rows $B$ and $D$ receive the same practice and development allowances. For the success rates in Table S15, the interface gain is $C - A$, language-only practice gain is $B - A$, and InteractVLA practice gain is $D - C$. The interaction is

$$\Delta_{\text{interaction}} = (D - C) - (B - A). \tag{G.1}$$

Paired bootstrap replicates compute this contrast directly, retaining the shared tasks, trials, and practice-run indices. Practice tokens and evaluation tokens occupy separate columns so the contrast can be inspected alongside its interaction budget.

## G.4. Learning Curves and Memory Ablations

The main-paper historical curve measures *cumulative case coverage*. For its fixed cohort $\mathcal{C}$ of 505 recorded cases,

$$\text{Coverage}(a) = \frac{100}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \mathbb{1}[\exists b \le a : \; b \text{ validly solves } c]\,. \tag{G.2}$$

Cases contribute once, including those first attempted after the displayed starting point. Attempts are ordered by completion timestamp, with tied timestamps grouped. The displayed interval runs from 131 solved cases to 244 solved cases over 1,519 additional attempts. Invalid attempts consume execution resources and preserve the prior coverage count. This statistic follows discovery of successful case solutions; the frozen-snapshot protocol below measures transfer to new instances.

After 0, 1, 2, 4, 6, 8, and 10 practice attempts per case, the reference learning-curve study freezes a strategy snapshot and evaluates it on the same disjoint evaluation seeds. Snapshot evaluation generates analysis records without updating strategy state. Curves report success against practice attempts, executed environment steps, agent tokens, and elapsed practice time; their uncertainty retains the five independent practice runs. The memory study uses the same snapshots and budget, changing the persistent content or retrieval access as listed in Table S16. In the memory-removal row, the final strategy prompt is restored to its initial version and retrieval is disabled; VLA parameters and generic tools stay fixed.

**Revision trace and rollback.** A complete trace includes an unsuccessful candidate. For example, the diagnostic case "increase the pre-contact offset after an obstructed approach" triggers a candidate target rule, which is evaluated against the parent on the five development scenes. If the larger offset reduces completed grasps, the acceptance rule retains the parent and records the obstruction hypothesis, offset change, failed seeds, and extra cost. The next revision can condition the offset on visible clearance. This trace format links each retained rule to its validating episodes and preserves failed hypotheses for subsequent retrieval.

Table S14. Reference information and experience boundaries. Scene-instance keys include task, asset, kitchen or layout, and initialization seed. Evaluation accesses the selected memory snapshot in read-only mode. Composition-transfer experiments additionally reserve task-semantic keys from practice.

| Information | VLA adaptation | RSI practice | Development selection | Final evaluation |
|---|---|---|---|---|
| Task semantics | Checkpoint data inventory | Practiced-task partition | Same practiced semantics | Practiced / held-out composition |
| Objects and scene instances | Training manifest | Practice manifest | Disjoint development IDs | Disjoint evaluation IDs |
| Random seeds | Training namespace | Practice namespace | Development namespace | Evaluation namespace |
| Demonstrations / poses | Training access | Reference-pose control only | Same condition-specific access | Same condition-specific access |
| Task compositions | Checkpoint inventory | Practice compositions | Practice compositions | Instance / composition split |
| Success signals | Training annotations | Terminal evaluator outcome | Terminal evaluator outcome | Scoring log after termination |
| Persistent memory | Fixed checkpoint | Versioned read / write | Candidate comparison | Selected snapshot: read only |

Table S15. Interface–RSI factorial design with Astra and MR-0 on RoboDojo. Practice cost is charged once per case; evaluation tokens are reported separately. The interaction is $(D - C) - (B - A)$. Mean ± SD is over five independent runs.

| ID | Interface | RSI | SR (%) | 95% CI | Practice M | Evaluation M |
|---|---|---|---|---|---|---|
| A | Language-only | No | $25.7 \pm 3.2$ | [22.5, 28.7] | 0.00 | $41.28 \pm 1.36$ |
| B | Language-only | Yes | $31.6 \pm 1.9$ | [29.1, 34.0] | 210.00 | $36.45 \pm 2.12$ |
| C | InteractVLA | No | $35.0 \pm 2.5$ | [32.1, 37.6] | 0.00 | $8.37 \pm 0.59$ |
| D | InteractVLA | Yes | $48.6 \pm 1.9$ | [47.3, 50.0] | 29.00 | $6.51 \pm 0.40$ |

# H. Controlled Comparisons and Component Ablations

## H.1. Closest Steering Methods and Output-Space Correction

The steering study fixes Astra, MR-0, the RoboDojo manifest, and agent observations and tools. VLS generates differentiable staged rewards from visual-language inputs and combines gradient steering with particle initialization, repulsion, Feynman–Kac resampling, adaptive guidance, and stage switching [5]. The reference adapter follows Sec. IV of version 1, including one refinement step for its flow-policy branch. A full-pipeline comparison retains generated rewards and stage management. A matched-target comparison supplies both samplers the same complete-pose target, controller mapping, and scalar soft-time objective; the VLS-target adapter retains its particle and resampling mechanism. The adapter version and changes are saved with the run configuration.

VLS particle counts are selected from $\{1, 4, 8\}$ under the shared development allowance. InteractVLA uses one trajectory, ten Euler steps, gain 1, correction cap 0.3, and the active interval $[0.5, 0.8)$. Two budget controls accompany the equal-episode comparison: equal measured GPU time per call, and equal total forward/backward work counted over all particles. Particle batching and resampling overhead are included. For the full-pipeline control, each method constructs its own target or reward from the same observations; the matched-target control caches target streams generated by the same agent and reuses them at paired states. Table S17 uses the matched-target setting.

**Post-generation trajectory optimization.** Given an unguided chunk $A^0$, the output-space baseline solves

$$\min_{A \in \mathcal{U}} \quad F_{t_*}(E(\mathcal{C}_{o_k}(A), G)) + \frac{\lambda_f}{H}\|D^{-1}(A - A^0)\|_F^2 + \frac{\lambda_s}{\max(H-2,1)}\|\Delta^2 D^{-1}(A - A^0)\|_F^2, \quad \text{(H.1)}$$

where $F_{t_*}$ is the soft minimum in Eq. (B.3) evaluated at the last active generation time $t_* = 0.7$, $E$ is the vector of complete-pose errors, and $\mathcal{U}$ is the command-bound set. Norms include valid channels only, with $D = I$ in the normalized action representation. The controller map $\mathcal{C}_{o_k}$ is shared with guidance. Projected Adam uses ten iterations, learning rate 0.05, $\lambda_f = 0.1$, and $\lambda_s = 0.01$; development selection has the same configuration allowance as other methods. Each iterate projects commands into $\mathcal{U}$. Agent local editing uses the bounded edit tool in Section D. The identity-Jacobian condition substitutes $I$ for the Jacobian of the clean-action estimate with respect to the current action latent, while retaining the current proposal, target objective, controller mapping, gradient cap, and denoising schedule.

Pose error is the positional distance at the jointly selected arrival index, accompanied in the result ledger by rotation error in degrees and gripper error. Motion quality is the RMS third finite difference of the executed end-effector positions divided by the native control timestep cubed, in $\mathrm{m/s^3}$. Reports also include collision and command-clipping rates. These episode metrics are aggregated within each run before reporting the five-run mean and SD. The evaluator measures GPU time and end-to-end episode time separately, including rejected proposals and recovery.

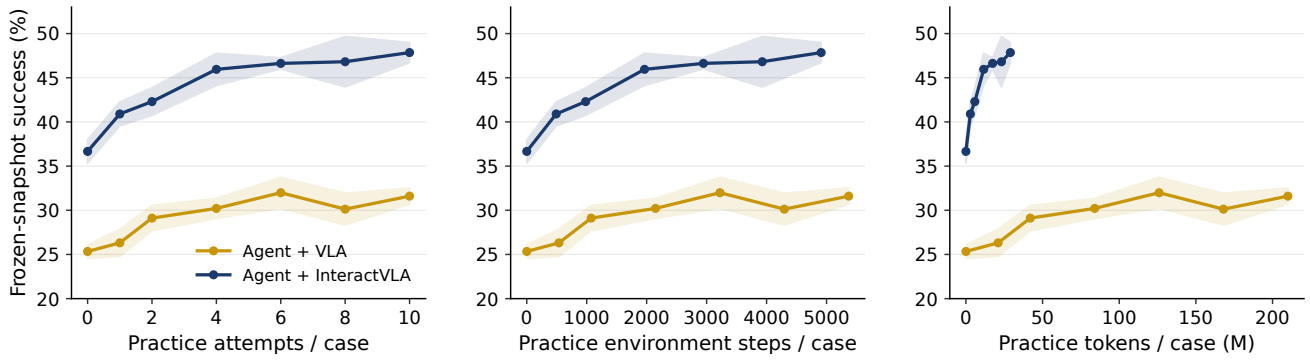



Figure S4. **RSI progress under explicit practice budgets.** A strategy snapshot is fixed at each checkpoint and evaluated on held-out instances. The panels relate success to practice attempts, environment steps, and agent tokens, while retaining the same evaluation task weights. Lines show the mean across five independent practice runs, and shading shows one sample standard deviation.

Table S16. Memory content under matched practice interactions. The three success columns evaluate new scenes of practiced tasks, held-out object instances, and held-out task compositions, respectively. Mean ± SD is over five independent runs.

| Memory condition | New scenes | New objects | New compositions | Tokens (M) |
|---|---|---|---|---|
| Repeated attempts, no memory | $35.9 \pm 1.5$ | $31.1 \pm 2.7$ | $24.8 \pm 3.0$ | $38.70 \pm 2.66$ |
| Successful examples | $41.7 \pm 2.5$ | $36.4 \pm 2.3$ | $27.7 \pm 2.3$ | $36.39 \pm 2.14$ |
| Success and failure experience | $45.2 \pm 3.3$ | $40.0 \pm 3.2$ | $33.3 \pm 3.8$ | $35.75 \pm 1.85$ |
| Full strategy update | $48.6 \pm 1.9$ | $45.0 \pm 2.4$ | $37.9 \pm 2.6$ | $34.85 \pm 1.72$ |
| Full update, memory removed | $35.5 \pm 2.1$ | $32.4 \pm 2.3$ | $27.0 \pm 3.9$ | $36.65 \pm 4.39$ |

### H.2. Forward Guidance and Reverse Feedback

The $2 \times 2$ comparison crosses geometric guidance with action-informed feedback. Every row retains calibrated observations and identical robot-control permissions. Feedback exposes the proposed chunk, its executed prefix, unexecuted continuation, and reprojection; without feedback, the agent receives ordinary observations and tool-status messages. The unguided-feedback condition exposes the raw policy proposal. Target generation uses the same agent schema in every row, with the target field ignored by the language-only sampler. Table S18 records success, agent requests, and total time, and its paired difference-in-differences measures the interaction between guidance and feedback.

### H.3. Feedback Content and Recovery Routes

Table S19 contains two controlled groups. The first fixes guidance, both recovery routes, and an enabled compatibility gate, and varies feedback from observations, to observations plus action chunks, to those inputs plus calibrated reprojection. Its three success-rate rows preserve the corresponding main-paper feedback comparison. The second fixes full feedback and guidance, disables the gate, and permits target revision, local editing, or both. Its final row enables the gate in the both-routes condition. Thus the upper group's "+ Reprojection" and the lower group's "Both + compatibility gate" are the same complete configuration and share their reported values. All rows use the same practice budget and generic direct-control tools.

A recovery event begins when the agent declares a discrepancy or receives a rejected call and ends at the next subgoal completion, abandonment, or episode termination. The ledger records route selection, target displacement, edited segment length, calls, time, and final outcome. Conditional recovery success divides successful recoveries by all selected events for that route. Route frequencies use all recovery events as the denominator; a later switch to another route is recorded as a new event linked to its parent. This accounting connects feedback content to the decisions it changes.

### H.4. Target Source and Reference-Pose Controls

Five RoboDojo conditions separate target quality from the steering mechanism. Training-data reference poses come from task-matched successful demonstrations: contact and gripper transitions identify interaction events, and the corresponding end-effector pose is transformed into the declared robot-base frame. The reference remains fixed under an evaluation perturbation. Fixed offline targets are generated once

Table S17. Steering comparisons with the same online agent and targets. Pose error uses measured target arrival; jerk is in $\mathrm{m/s^3}$. GPU time is per VLA call; wall time includes the entire evaluation episode. All methods have the same rollout budget. Mean $\pm$ SD is over five independent runs.

| Method | SR (%) | Error (cm) | Jerk | GPU (ms) | Wall (s) |
|---|---|---|---|---|---|
| Language-only Agent + VLA | $26.9 \pm 1.5$ | $7.66 \pm 0.30$ | $23.5 \pm 1.9$ | $313 \pm 10$ | $216 \pm 10$ |
| Local action editing | $29.4 \pm 2.3$ | $3.85 \pm 0.18$ | $39.3 \pm 2.3$ | $308 \pm 17$ | $238 \pm 5$ |
| Post-generation optimization | $35.3 \pm 3.0$ | $2.23 \pm 0.08$ | $35.2 \pm 1.0$ | $434 \pm 18$ | $232 \pm 12$ |
| VLS adaptation (4 particles) | $33.6 \pm 2.4$ | $2.05 \pm 0.12$ | $30.0 \pm 1.8$ | $1415 \pm 92$ | $281 \pm 20$ |
| Identity-Jacobian guidance | $34.5 \pm 2.6$ | $2.44 \pm 0.18$ | $30.6 \pm 1.5$ | $352 \pm 17$ | $213 \pm 15$ |
| InteractVLA | $38.4 \pm 3.3$ | $1.68 \pm 0.09$ | $26.5 \pm 1.6$ | $535 \pm 39$ | $199 \pm 6$ |

Table S18. Forward and reverse interface components on RoboDojo, with fixed robot-control permissions. Calls and wall time are per evaluation episode. Mean $\pm$ SD is over five independent runs.

| Guidance | Feedback | SR (%) | Calls | Time (s) |
|---|---|---|---|---|
| No | No | $25.4 \pm 2.5$ | $15.7 \pm 1.1$ | $211 \pm 10$ |
| Yes | No | $29.9 \pm 2.1$ | $12.4 \pm 0.4$ | $212 \pm 15$ |
| No | Yes | $28.4 \pm 2.0$ | $14.2 \pm 1.3$ | $212 \pm 3$ |
| Yes | Yes | $35.7 \pm 1.6$ | $9.2 \pm 0.3$ | $195 \pm 14$ |

from the initial policy-visible observation with the shared agent; online targets are regenerated after the execution prefix or a rejection. The current-scene oracle obtains an interaction pose from the simulator task's object geometry and diagnostic subgoal annotation, and is evaluated as a separate information condition.

Target error uses the distance to a valid current-scene interaction-pose set, together with orientation and gripper discrepancy in the log. Its pairing with rejection and success rates measures how grounding quality affects usable guidance. A companion LIBERO-Plus experiment repeats the robot-initialization reference-pose condition and defines its fallback operationally: the unguided command is retained when the reference target fails the observable workspace, transform-validity, or compatibility checks. The implementation logs the triggering check before applying the fallback; benchmark perturbation labels are reserved for analysis. Oracle and agent rows share the same sampler and budgets, allowing the oracle–agent gap to localize target-selection effects.

# I. Generalization Across Agent Models and VLA Backbones

## I.1. Reduced-Repeat Evaluation Protocol

Every cross-model configuration evaluates **all official RoboDojo tasks with exactly 20 trials per task in total**. Five independent practice chains each receive four disjoint evaluation seeds per task. Their task-weighted scores supply the reported mean and SD; all methods use the same five seed partitions. Relative to the 50-trial reference protocol, this uses $20/50 = 40\%$ of the repetition budget while retaining complete task coverage. Task coverage and repetition count are recorded as separate manifest fields. The paired comparison fixes episode and API limits, observations, generic tools, and the development-selection allowance. Each agent–backbone pair independently performs the same ten-attempt practice protocol in five runs and then fixes its memory for evaluation. The two interface variants share the checkpoint, adaptation history, action-noise seeds, and provider settings; the policy-tool interface is the changed factor.

The reported interface effect is

$$\Delta_{\mathrm{SR}} = \mathrm{SR}_{\mathrm{Agent+InteractVLA}} - \mathrm{SR}_{\mathrm{Agent+VLA}}, \qquad \text{(I.1)}$$

in percentage points, with the mean and sample SD of the five paired run-level gains and additional bootstrap intervals from Section F. The result ledger includes all per-task success counts and the Generalization, Precision, Long-Horizon, Memory, and Open summaries. Absolute success describes the complete configuration; the within-pair contrast measures its benefit from the interface.

## I.2. Cross-Agent Results

The agent axis fixes MR-0 and compares GPT-6.1 Sol, GPT-6 Astra, and Claude Fable 5.2. These human-readable labels are mapped to the actual provider-returned model identifiers, endpoint versions, decoding settings, context limits, and response-usage fields in each run manifest. The same

Table S19. Feedback and recovery ablations. The upper group varies feedback with both recovery routes and the compatibility gate enabled. The lower group fixes full feedback and varies recovery routes; only its final row enables the gate. Calls and time summarize RoboDojo recovery episodes. Mean ± SD is over five independent runs.

| Variant | LP | Dojo | Casa | Calls | Time (s) |
|---|---|---|---|---|---|
| Observations only | $97.9 \pm 0.2$ | $40.3 \pm 3.6$ | $84.1 \pm 0.4$ | $6.4 \pm 0.4$ | $241 \pm 9$ |
| + Generated actions | $98.7 \pm 0.1$ | $44.8 \pm 2.6$ | $87.0 \pm 0.6$ | $4.4 \pm 0.1$ | $226 \pm 16$ |
| + Reprojection | $99.4 \pm 0.2$ | $48.6 \pm 1.9$ | $89.4 \pm 0.9$ | $2.9 \pm 0.1$ | $190 \pm 10$ |
| Target revision only | $97.8 \pm 0.5$ | $43.3 \pm 2.8$ | $84.8 \pm 2.0$ | $4.9 \pm 0.5$ | $214 \pm 14$ |
| Local edits only | $97.5 \pm 0.4$ | $41.8 \pm 2.2$ | $82.4 \pm 1.6$ | $5.5 \pm 0.1$ | $227 \pm 15$ |
| Both recovery routes | $98.8 \pm 0.1$ | $47.2 \pm 1.4$ | $88.1 \pm 1.2$ | $3.4 \pm 0.2$ | $203 \pm 15$ |
| Both + compatibility gate | $99.4 \pm 0.2$ | $48.6 \pm 1.9$ | $89.4 \pm 0.9$ | $2.9 \pm 0.1$ | $190 \pm 10$ |

Table S20. Target information conditions on RoboDojo. Target error is evaluated before guidance; rejection is per requested call. The oracle row has privileged simulator geometry. Mean ± SD is over five independent runs.

| Target source | Error (cm) | SR (%) | Reject (%) | Calls | Time (s) |
|---|---|---|---|---|---|
| No geometric target | — | $24.0 \pm 2.1$ | 0.0 | $15.2 \pm 0.8$ | $219 \pm 10$ |
| Training-data reference | $6.35 \pm 0.50$ | $28.1 \pm 1.5$ | 18.0 | $14.2 \pm 0.8$ | $217 \pm 11$ |
| Fixed offline agent | $3.45 \pm 0.28$ | $31.4 \pm 2.7$ | 14.0 | $12.1 \pm 0.4$ | $214 \pm 11$ |
| Online agent | $2.06 \pm 0.19$ | $35.9 \pm 2.1$ | 10.0 | $9.3 \pm 0.5$ | $201 \pm 18$ |
| Current-scene oracle | $0.30 \pm 0.01$ | $43.6 \pm 4.0$ | 4.0 | $7.9 \pm 0.7$ | $181 \pm 9$ |

prompt content, visual inputs, and tool schemas pass through provider-specific serialization. Table S21 measures both absolute capability and the paired interface gain for each agent; pairwise agent differences use the same held-out scenes and task weights.

### I.3. Cross-Backbone Results

The backbone axis fixes Fable 5.2 and compares MR-0 with $\pi_{0.5}$. Each backbone retains its native observation encoder, action normalization, action representation, and corresponding command-to-pose map. Within a backbone, both interfaces load the same checkpoint and data-adaptation history. The manifest stores the checkpoint digest, dataset inventory, optimizer and adaptation budget where applicable, native chunk length, command bounds, and Euler schedule. Each backbone receives the same development-tuning allowance for guidance, and the selected setting is fixed across tasks.

The MR-0/Fable row is shared between the cross-agent and cross-backbone tables through the same run IDs. This makes the repeated row a single configuration with identical seeds and memory snapshots. Category-level gains, target residuals, rejection rates, and mapping checks accompany the overall comparison. These diagnostics locate a lower starting capability, a target-grounding error, or a controller-mapping discrepancy within the paired experiment. The study varies one model axis at a time and reports the exact evaluated combinations in its manifest.

## J. Efficiency, Compute, and Cost Accounting

### J.1. Accounting Boundaries

The main efficiency figure compares **Agent + VLA**, which issues language-only policy calls; **Agent + InteractVLA**, which additionally specifies geometric targets and inspects or refines policy proposals; and **Agent Only**, which directly controls robot motion. A case is one task-instance problem together with *all of its RSI attempts*. Each case's tokens, executed control steps, and elapsed time are summed over its entire attempt history, including failures and retries, and the figure reports their mean across recorded cases. Its success rates belong to these efficiency-comparison configurations. The cost denominator for every normalized panel is Agent + VLA on the same benchmark.

For the controlled protocol, costs additionally have mutually exclusive practice, validation, and fixed-strategy evaluation tags. Practice includes rollout reasoning, memory retrieval and indexing, experience summarization, and prompt or strategy revisions. Candidate comparisons enter validation, including rejected updates. Evaluation includes rejected calls, recovery, unsuccessful actions, and final termination. Given a selected strategy deployed for $N$ episodes, the amor-

Table S21. Cross-agent design with MR-0 and full RSI. Every official task receives 20 paired trials in total: four evaluation seeds in each of five independent practice runs; the agent varies while checkpoint, practice budget, and tools remain fixed. Mean ± SD is over five independent runs.

| Agent | Agent + VLA | Agent + InteractVLA | Δ (pp) | Paired 95% CI |
|---|---|---|---|---|
| GPT-6.1 Sol | $17.6 \pm 3.0$ | $29.8 \pm 2.1$ | $12.2 \pm 2.3$ | [9.3, 15.1] |
| GPT-6 Astra | $30.4 \pm 2.8$ | $49.4 \pm 3.1$ | $19.0 \pm 0.6$ | [16.3, 21.6] |
| Claude Fable 5.2 | $53.8 \pm 3.6$ | $72.2 \pm 2.9$ | $18.4 \pm 2.6$ | [15.1, 21.9] |

Table S22. Cross-backbone design with Claude Fable 5.2. Each paired interface comparison shares one checkpoint, adaptation history, practice budget, and task-seed manifest. The MR-0 row reuses the identical cross-agent records. Mean ± SD is over five independent runs.

| Backbone | Agent + VLA | Agent + InteractVLA | Δ (pp) | Paired 95% CI |
|---|---|---|---|---|
| MR-0 | $53.8 \pm 3.6$ | $72.2 \pm 2.9$ | $18.4 \pm 2.6$ | [15.1, 21.9] |
| $\pi_{0.5}$ | $45.4 \pm 4.6$ | $65.0 \pm 2.7$ | $19.6 \pm 3.8$ | [16.1, 23.8] |

tized cost is

$$C_{\text{amortized}}(N) = \frac{C_{\text{practice}} + C_{\text{validation}}}{N} + \frac{1}{N}\sum_{i=1}^{N} C_{\text{eval},i}. \tag{J.1}$$

Historical cumulative-case accounting and this deployment amortization occupy separate fields, preserving the measured scope of the main efficiency data.

**Time decomposition.** The end-to-end timer begins before the first agent or policy request and ends after the terminal environment outcome and required strategy update. It includes API wait, policy inference and gradient computation, environment execution, and orchestration. GPU regions use synchronized events at measurement boundaries; CPU regions use a monotonic clock. Request, policy, projection, and controller intervals have timestamps and parent spans. Their union defines elapsed wall time, while interval durations define component work. The serial reference runner awaits each command before issuing the next; asynchronous variants additionally record overlap so the component-work total and elapsed total retain distinct meanings.

**Tokens and API use.** The usage ledger stores provider-reported input, output, cached-input, image, and reasoning fields, together with request ID and billing status. A total includes each provider usage category once according to its schema. Agent requests, VLA chunks, direct control requests, and executed controller steps have separate counters. A direct-control request that executes a multi-step segment contributes one tool request and the actual number of executed steps. API requests needed to formulate that request are counted independently. A monetary view multiplies billed usage by the provider's dated price record, including cache and retry treatment; the token and time comparisons themselves require no assumed price.

## J.2. Absolute Cost Tables

Table S23 contains the absolute cumulative costs underlying the main figure, including the full unsuccessful attempt history. Astra is used on RoboDojo and RoboCasa365, and Sol on LIBERO-Plus. The component ledger in Table S24 separates token direction, API frequency, and elapsed work categories, and Table S25 isolates the cost of an individual policy request.

**Per-call measurement.** The reference microbenchmark uses one NVIDIA A100 80-GB GPU, batch size one, bfloat16 model inference, and float32 geometric objectives, with RoboDojo chunks of length 50. After 20 warm-up requests, 200 seeded requests supply mean and sample standard deviation for latency, allocated peak GPU memory, and projection time; the event ledger also retains the median and 5th–95th percentiles. Both unguided and accepted guided calls use ten forward evaluations; the active interval contributes three backward evaluations. A rejected request records the actual evaluations reached before termination. Agent latency is measured separately from the policy microbenchmark, using the same fixed observation payload and provider configuration. The hardware, software build, synchronization mode, and model digest accompany every timing record.

The main-data comparison measures the total effect of the complete agent strategy, while the microbenchmark measures incremental guidance computation per call. Linking the two requires the observed counts of policy requests, rejected calls, regenerated targets, direct-control requests, and

Table S23. Absolute costs supplied for the main efficiency comparison. Each case accumulates all RSI attempts, including failed attempts, and the table reports the mean over cases. Time is end-to-end, including agent API latency, VLA inference and guidance, and environment execution.

| Benchmark | Configuration | SR (%) | Steps | Tokens (M) | Time (min) |
|---|---|---|---|---|---|
| RoboDojo | Agent + VLA | 31.70 | 6258 | 247.000 | 157.0 |
| | Agent + InteractVLA | 48.62 | 6064 | 35.430 | 91.0 |
| | Agent Only | 39.60 | 5139 | 788.000 | 2022.0 |
| RoboCasa365 | Agent + VLA | 79.50 | 2142 | 9.520 | 17.7 |
| | Agent + InteractVLA | 89.40 | 1730 | 4.200 | 10.8 |
| | Agent Only | 61.50 | 1603 | 225.000 | 576.0 |
| LIBERO-Plus | Agent + VLA | 96.40 | 241 | 0.219 | 2.6 |
| | Agent + InteractVLA | 99.40 | 225 | 0.196 | 0.5 |
| | Agent Only | 95.10 | 204 | 29.200 | 75.0 |

Table S24. Decomposition of the historical RoboDojo cumulative-case comparison, separate from the prospective fixed-budget protocol. Token columns are millions; four disjoint wall-time columns are minutes. API includes remote waiting, VLA includes guidance, and Other includes orchestration. The component sums equal the supplied end-to-end totals.

| Configuration | Input M | Output M | API calls | API | VLA | Env. | Other |
|---|---|---|---|---|---|---|---|
| Agent + VLA | 224.77 | 22.23 | 1720 | 118.0 | 8.0 | 27.0 | 4.0 |
| Agent + InteractVLA | 31.18 | 4.25 | 265 | 57.0 | 12.0 | 19.0 | 3.0 |
| Agent Only | 740.72 | 47.28 | 5139 | 1920.0 | 0.0 | 92.0 | 10.0 |

agent APIs. Their joint ledger tests whether fewer high-level decisions and recovery attempts offset the cost of differentiating through the action expert.

### J.3. Budget-Matched Performance

Budget curves use separate token, executed-step, and elapsed-time caps. Their checkpoints are fixed before evaluation at $\{0.25, 0.5, 1, 2, 4\}$ times the language-only development-set median for that resource, and the resulting absolute budgets are saved. A run stops at success, its episode limit, or the first exhausted budget. Token and step admission uses the request's maximum output and execution-prefix limits. Wall-clock admission uses the remaining deadline and cancels pending work at expiration. Work already issued at the boundary is charged in full, and the report stores any overshoot. Unsuccessful runs remain in the success denominator and consume their full recorded cost.

Frozen-strategy curves evaluate deployment efficiency without updating memory, and practice curves evaluate the evolving strategy at fixed snapshot checkpoints. Equal-budget success is computed at the registered caps. Cost-to-success uses a common threshold selected from the range attained by all compared methods, with unsuccessful threshold crossings treated as right-censored at the budget limit. Bootstrap intervals preserve the paired scenes and practice runs. Within each benchmark, all three efficiency conditions use the same agent model, observation encoding, and provider settings.

## K. Real-World Platform, Scoring, and Detailed Results

### K.1. Hardware, Demonstrations, and Training

The Dexmal evaluation comprises Tabletop Tidying, Upright Bottle, Water Pouring, and Move Cube, with 50 demonstrations per task. Standalone VLA, language-only Agent + VLA, and Agent + InteractVLA share one four-task MR-0 checkpoint. The reference configuration uses position-controlled arms at 10 Hz, calibrated gripper commands, and RGB-D images at $640 \times 480$ and 30 Hz. A fixed scene camera and available wrist cameras provide timestamped observations. A hardware manifest records actual arm and gripper models, camera serials and poses, intrinsics, depth scale, controller gains, velocity and force limits, compute devices, and software versions. Calibration uses the same camera-to-base transforms for grounding, projection, and scoring, with reprojection residuals stored alongside the transforms.

The reference adaptation splits each task's demonstrations into 40 training and ten validation trajectories by scene identity. Demonstrations cover object placement, initial orienta-

Table S25. Call-level instrumentation at batch size one, BF16, $H = 50$, on one 80-GB A100. Timing and peak memory are mean $\pm$ sample SD over 200 requests per configuration, following 20 warm-up requests. Forward/backward counts are exact. API latency describes one planning request separately from GPU call time; rejected calls omit the trajectory overlay.

| Configuration | Fwd | Bwd | GPU (ms) | Peak (GiB) | API (s) | Project (ms) |
|---|---|---|---|---|---|---|
| MR-0, unguided | 10 | 0 | $310.7 \pm 19.5$ | $11.80 \pm 0.13$ | $12.54 \pm 3.19$ | $1.81 \pm 0.22$ |
| MR-0, guided | 10 | 3 | $509.8 \pm 32.1$ | $16.90 \pm 0.17$ | $12.98 \pm 3.10$ | $1.90 \pm 0.21$ |
| $\pi_{0.5}$, unguided | 10 | 0 | $288.8 \pm 19.5$ | $10.61 \pm 0.11$ | $12.37 \pm 3.03$ | $1.82 \pm 0.23$ |
| $\pi_{0.5}$, guided | 10 | 3 | $471.8 \pm 28.0$ | $15.70 \pm 0.15$ | $12.76 \pm 3.20$ | $1.91 \pm 0.21$ |
| MR-0, rejected at $t = 0.6$ | 7 | 2 | $347.4 \pm 19.5$ | $16.90 \pm 0.16$ | $12.47 \pm 3.12$ | $0.00 \pm 0.00$ |

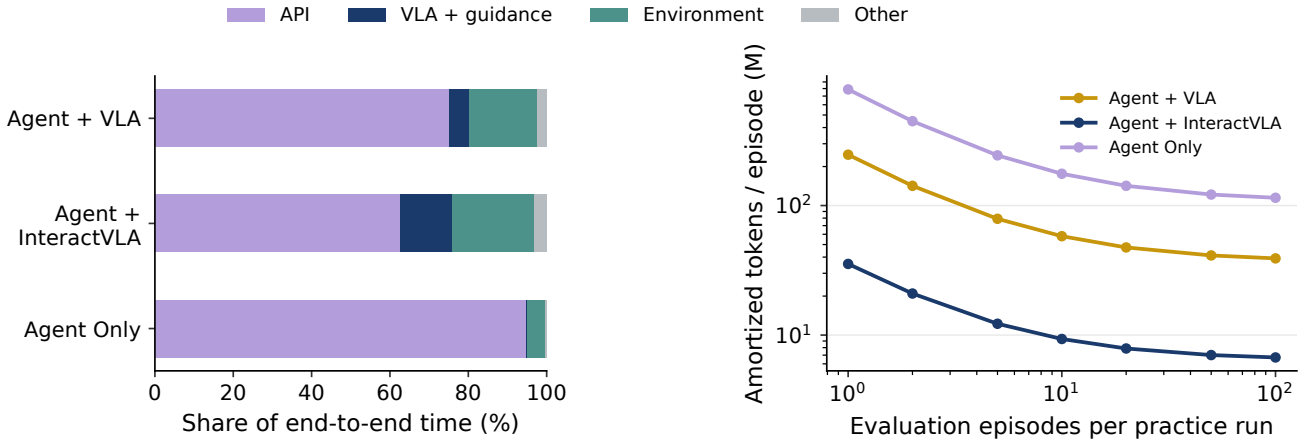



Figure S5. **End-to-end cost composition and practice amortization.** Left: fractions of total time assigned to agent API waiting, policy inference and guidance, environment execution, and other orchestration. Right: amortized agent tokens per deployment episode as the number of episodes sharing one acquired strategy increases. The practice cost is paid once and combined with the mean fixed-strategy evaluation cost.

tion, approach direction, and grasp location. Task-balanced sampling trains the action expert and its input projection for 10,000 AdamW steps, using batch size 32, learning rate $2 \times 10^{-5}$, weight decay 0.01, 500 warm-up steps, and cosine decay to $2 \times 10^{-6}$. Brightness and contrast augmentation use factors in $[0.9, 1.1]$; spatial calibration and action labels retain their original coordinates. Validation denoising loss, evaluated every 500 steps, selects the shared checkpoint. The checkpoint manifest identifies the original pretrained weights and all 200 adaptation demonstrations.

## K.2. Task Definitions and Initial-State Distributions

Table S26 defines the reference scoring protocol. Objects are sampled within the calibrated reachable tabletop polygon with nonoverlapping footprints and at least 2 cm clearance. The 30 initial configurations for each task fix object identities, positions, orientations, and relevant liquid quantities before method execution. Every trial has a 600-second limit.

Basket containment requires the object centroid inside the basket opening and the object's support below its rim. Bottle orientation is measured from its fitted principal axis. Cup and spill-tray mass changes use tared scales. Water Pouring assigns score one to complete success; otherwise its score is

$$r = \min\left\{0.99, \max\left[0, \frac{m_{\text{cup}}}{80\,\text{g}} - \frac{m_{\text{spill}}}{20\,\text{g}}\right]\right\}. \qquad \text{(K.1)}$$

For Move Cube, grasp requires controlled retention and lift requires 3 cm table clearance. The 0.50 placement component awards 0.25 for containment and another 0.25 for stable release. Two reviewers score method-anonymized endpoint recordings using these rules; disagreements are resolved from the full trajectory and sensor log. Their initial labels and adjudicated outcome are preserved.

During real-world evaluation, various issues may arise, including violations of robot arm limits, discontinuous motions, and awkward grasping or placement angles. Operators apply score deductions at their discretion; these issues are not discussed in detail here. Some trials involving arm-limit

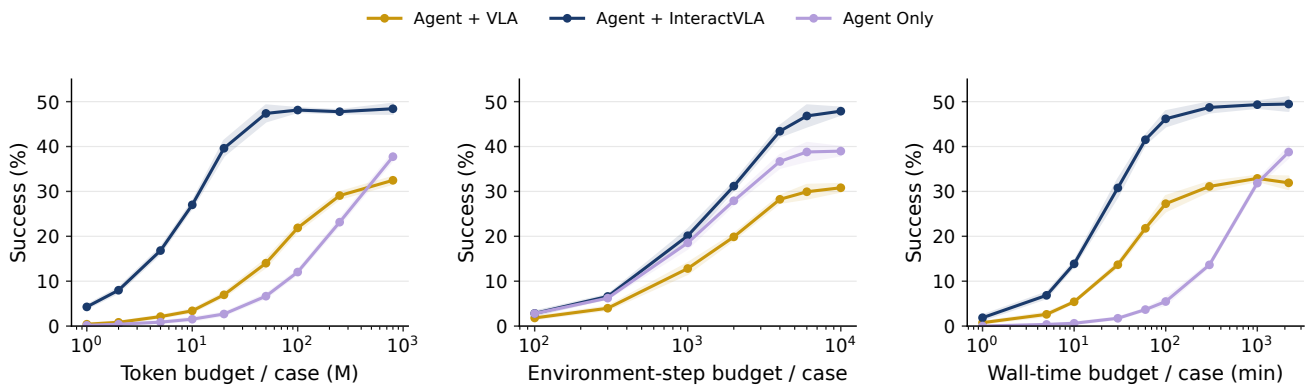



Figure S6. **Success under matched resource budgets.** Agent + VLA, Agent + InteractVLA, and Agent Only are compared at fixed token, controller-step, and end-to-end time budgets. All attempts consume budget. Frozen-strategy evaluation uses a fixed memory snapshot, while a separate practice ledger records the cost of acquiring that snapshot.

Table S26. Dexmal task definitions. Stability means satisfying the stated condition continuously for two seconds after release. Scores are normalized to $[0, 1]$ before conversion to percent.

| Task | Initial configuration | Complete success | Partial score |
|---|---|---|---|
| Tabletop Tidying | Four objects outside a basket; varied positions and yaw. | All four objects released inside the basket and stable. | Number of correctly placed objects divided by four. |
| Upright Bottle | Three bottles initially lying on their sides, with varied position and yaw. | All bottle axes within 15° of vertical, supported and stable. | Number of upright, stable bottles divided by three. |
| Water Pouring | Source bottle containing 100 g water; empty receiving cup and spill tray. | At least 80 g transferred, at most 5 g spilled, bottle restored upright and stable. | Transfer fraction with spill penalty, as defined below. |
| Move Cube | A 5.7 cm cube and cup with inner diameter at least 8.5 cm. | Cube contained in cup, released and stable. | Grasp, lift, and stable placement receive 0.25, 0.25, and 0.50. |

violations or emergency stops are included in the reported results with reduced weights.

### K.3. Success Rate, Task Score, and Standard Deviation

For 30 trials, binary completion $s_j$ and partial score $r_j \in [0, 1]$ produce

$$\mathrm{SR} = \frac{100}{30} \sum_j s_j, \qquad \overline{R} = \frac{100}{30} \sum_j r_j. \tag{K.2}$$

Score, elapsed time, and API calls use the **mean ± sample standard deviation** reporting format, with the 30-trial protocol defining $\mathrm{SD}(y) = \sqrt{\sum_j (y_j - \bar{y})^2 / 29}$. The SD describes trial variation; a confidence interval describes uncertainty in a population quantity. Table S27 preserves the supplied mean task scores and costs. Tables S28 and S29 expand those same mean values with the supplementary variability summaries. Their Score and time means match the main-paper real-world table, and their API-call means retain the corresponding supplied values. Numerical sources for the means and variability summaries are recorded separately in the final declaration.

Table S30 separately specifies the binary-completion diagnostic under the 30-trial protocol: successful-trial counts, complete-task SR, Wilson 95% intervals, and stops. Its completion counts belong to that diagnostic ledger; the main-paper Score remains the fractional task metric. The simulation studies report five-run mean and SD according to Eq. (F.2).

### K.4. Timing, Pairing, and Exceptional Events

Timing starts when the first observation is released to the method and ends at verified completion, termination, or the 600-second limit. It includes agent waiting, policy generation, guidance, and robot execution. All trials enter the mean, with failures charged their actual elapsed time. A separately identified successful-trial statistic supports latency analysis conditional on completion.

Method order is randomized within each initial-configuration block. Reset verification compares object poses against the saved configuration, with 5 mm transla-

Table S27. Author-supplied Dexmal aggregates. Score is normalized task score; time and agent calls are episode averages. These values retain their supplied metric definition.

| Task | Method | Score (%) | Time (s) | API calls |
|---|---|---|---|---|
| Tabletop Tidying | VLA | 31.2 | 97.4 | 0.0 |
| | Agent + VLA | 65.5 | 372.0 | 23.5 |
| | Agent + InteractVLA | 92.5 | 295.8 | 9.9 |
| Upright Bottle | VLA | 16.9 | 63.8 | 0.0 |
| | Agent + VLA | 37.9 | 226.8 | 17.8 |
| | Agent + InteractVLA | 74.6 | 217.1 | 11.5 |
| Water Pouring | VLA | 41.6 | 27.4 | 0.0 |
| | Agent + VLA | 38.5 | 97.0 | 6.1 |
| | Agent + InteractVLA | 79.2 | 71.2 | 4.4 |
| Move Cube | VLA | 51.5 | 22.9 | 0.0 |
| | Agent + VLA | 86.7 | 165.2 | 14.4 |
| | Agent + InteractVLA | 97.6 | 67.5 | 4.7 |

Table S28. Dexmal task-score summary. Means follow the task scores in the main paper; each entry reports mean ± standard deviation. Completion counts are reported separately in Table S30.

| Task | VLA | Agent + VLA | Agent + InteractVLA |
|---|---|---|---|
| Tabletop Tidying | $31.2 \pm 16.2$ | $65.5 \pm 16.7$ | $92.5 \pm 8.3$ |
| Upright Bottle | $16.9 \pm 13.3$ | $37.9 \pm 16.3$ | $74.6 \pm 13.0$ |
| Water Pouring | $41.6 \pm 14.6$ | $38.5 \pm 14.8$ | $79.2 \pm 14.6$ |
| Move Cube | $51.5 \pm 17.5$ | $86.7 \pm 11.0$ | $97.6 \pm 4.6$ |

tion and 3° yaw tolerances; lighting, camera exposure, and robot state are logged. Operators reset between trials. An emergency stop or intervention during execution terminates the trial as failure and remains in the ledger. Comparisons report both standalone VLA and Agent + VLA denominators, since policy-only execution and agent-assisted execution have different decision costs.

### K.5. Real-World Recovery and Perturbation Cases

The recovery protocol introduces a 2 cm lateral object displacement after an approach prefix, a 0.5-second target-region occlusion before the next observation, or a target substitution before contact. Each condition is paired with an unperturbed configuration and records the trigger timestamp. Perturbation execution occurs while the robot is paused at a predefined boundary. Natural slipping receives an event label from object and gripper motion. A complete case record includes before/after observations, target poses, generated and executed actions, rejection or edit decisions, and the final scoring predicate. This record supports comparison of recovery frequency, success, time, and unrecovered failures against language-only Agent + VLA.

## L. Robustness, Motion Quality, and Failure Modes

### L.1. Target and Observation Perturbations

The RoboDojo robustness protocol changes one factor at a time at paired states in each of five independent runs. Position bias has uniformly sampled direction and magnitude 2 or 5 cm; orientation bias uses a random axis and angle 10° or 30°. Gripper bias is 10% or 30% of command range with random sign, while transition delay is two or five control steps. Depth perturbations use zero-mean Gaussian noise with standard deviation 5 or 20 mm, or independent missing samples at rates 10% or 40%. Calibration perturbation shifts camera extrinsic translation by 5 or 20 mm. Observation buffering delays target updates by 100 or 500 ms. Target-region masks obscure 10% or 40% of its pixels; object displacement shifts the interaction object by 2 or 5 cm after target generation. Both severity levels have equal weight, and the underlying valid target defines the error metric. Success and continuous episode metrics use run-level means and SDs.

### L.2. Geometric Accuracy Versus Task Completion

Target attainment uses the *measured* end-effector trajectory: one shared executed step must be within 2 cm in position, 15° in orientation, and 10% of gripper range. Commanded target residual and command-to-execution residual are logged separately. Table S32 and the failure taxonomy use a separate diagnostic cohort with shared trial IDs and binary outcomes. Their counts and fractions are computed directly from that cohort's ledger; the main-paper benchmark mean retains its own evaluation record. The joint outcome table crosses attainment with the task predicate, distinguish-

Table S29. Dexmal execution-time and agent-call summary, reported as mean ± standard deviation. Time includes the complete episode, and calls count agent API requests.

| Task | Method | Time (s) | API calls |
|---|---|---|---|
| Tabletop Tidying | VLA | $97.4 \pm 32.8$ | $0.0 \pm 0.0$ |
| | Agent + VLA | $372.0 \pm 111.0$ | $23.5 \pm 6.1$ |
| | Agent + InteractVLA | $295.8 \pm 77.7$ | $9.9 \pm 2.5$ |
| Upright Bottle | VLA | $63.8 \pm 20.1$ | $0.0 \pm 0.0$ |
| | Agent + VLA | $226.8 \pm 70.1$ | $17.8 \pm 4.3$ |
| | Agent + InteractVLA | $217.1 \pm 53.9$ | $11.5 \pm 3.6$ |
| Water Pouring | VLA | $27.4 \pm 6.9$ | $0.0 \pm 0.0$ |
| | Agent + VLA | $97.0 \pm 31.4$ | $6.1 \pm 1.6$ |
| | Agent + InteractVLA | $71.2 \pm 23.1$ | $4.4 \pm 1.3$ |
| Move Cube | VLA | $22.9 \pm 6.8$ | $0.0 \pm 0.0$ |
| | Agent + VLA | $165.2 \pm 57.4$ | $14.4 \pm 3.1$ |
| | Agent + InteractVLA | $67.5 \pm 22.3$ | $4.7 \pm 1.1$ |

Table S30. Complete-task outcomes and exceptional events under the 30-trial diagnostic protocol. Success counts and Wilson 95% intervals use the same trial denominator; human stops remain failures.

| Task | Method | Successes | SR (%) | 95% CI | Human stops |
|---|---|---|---|---|---|
| Tabletop Tidying | VLA | 5/30 | 16.7 | [7.3, 33.6] | 1 |
| | Agent + VLA | 13/30 | 43.3 | [27.4, 60.8] | 0 |
| | Agent + InteractVLA | 25/30 | 83.3 | [66.4, 92.7] | 0 |
| Upright Bottle | VLA | 3/30 | 10.0 | [3.5, 25.6] | 1 |
| | Agent + VLA | 8/30 | 26.7 | [14.2, 44.4] | 2 |
| | Agent + InteractVLA | 19/30 | 63.3 | [45.5, 78.1] | 0 |
| Water Pouring | VLA | 9/30 | 30.0 | [16.7, 47.9] | 3 |
| | Agent + VLA | 6/30 | 20.0 | [9.5, 37.3] | 0 |
| | Agent + InteractVLA | 20/30 | 66.7 | [48.8, 80.8] | 0 |
| Move Cube | VLA | 15/30 | 50.0 | [33.2, 66.8] | 0 |
| | Agent + VLA | 19/30 | 63.3 | [45.5, 78.1] | 2 |
| | Agent + InteractVLA | 28/30 | 93.3 | [78.7, 98.2] | 0 |

ing successful contact at an equivalent pose from failures after accurate arrival.

Path length is $\sum_h \|p_{h+1} - p_h\|_2$ in meters. Acceleration and jerk use second and third finite differences of measured position divided by the control timestep squared and cubed, followed by RMS aggregation. A saturated episode has at least one clipped control command; an undesired-contact episode activates the simulator's collision filter outside designated manipulation contacts. Proposal deviation is RMS difference in normalized valid action channels. Paired raw and guided proposals share observations and initial noise, and physical-motion metrics use the same executed duration and sampling rate.

### L.3. Category-Specific Regressions

The main online RoboDojo comparison motivates a dedicated Precision analysis with **GPT-6.1 Sol and MR-0**. The first two rows of Table S34 retain the main-paper Precision means of 9.7% for language-only calls and 8.3% for online InteractVLA. The additional diagnostic conditions replace online targets with current-scene geometric targets, use a controller-corrected mapping, and add contact-stage local re-

Table S31. One-factor perturbations on RoboDojo. Each pair of severities contributes equally. Arrival error uses the current, unperturbed target and measured end-effector state. Recovery cost counts additional agent calls after the first failed subgoal. Mean ± SD is over five independent runs.

| Perturbation | Severity pair | SR (%) | Error (cm) | Reject (%) | Recovery calls |
|---|---|---|---|---|---|
| No perturbation | 0 | $48.6 \pm 1.9$ | $1.74 \pm 0.15$ | 10.0 | $2.9 \pm 0.3$ |
| Position bias | 2 / 5 cm | $43.9 \pm 1.6$ | $3.06 \pm 0.16$ | 18.0 | $4.1 \pm 0.3$ |
| Orientation bias | 10 / 30 deg | $43.7 \pm 2.5$ | $2.43 \pm 0.18$ | 19.0 | $3.8 \pm 0.3$ |
| Gripper width bias | 10 / 30% range | $43.0 \pm 0.8$ | $1.91 \pm 0.12$ | 20.0 | $4.2 \pm 0.3$ |
| Gripper transition delay | 2 / 5 steps | $41.6 \pm 2.2$ | $2.33 \pm 0.10$ | 17.0 | $4.4 \pm 0.2$ |
| Depth noise | 5 / 20 mm | $46.8 \pm 0.8$ | $2.68 \pm 0.26$ | 15.0 | $3.5 \pm 0.2$ |
| Missing depth | 10 / 40% | $40.8 \pm 2.7$ | $3.17 \pm 0.24$ | 24.0 | $5.0 \pm 0.2$ |
| Calibration shift | 5 / 20 mm | $43.1 \pm 2.8$ | $2.59 \pm 0.20$ | 21.0 | $4.2 \pm 0.3$ |
| Target update delay | 100 / 500 ms | $40.0 \pm 2.2$ | $3.84 \pm 0.14$ | 23.0 | $5.4 \pm 0.2$ |
| Occluded target pixels | 10 / 40% | $38.4 \pm 2.7$ | $4.10 \pm 0.29$ | 26.0 | $5.4 \pm 0.4$ |
| Object displacement | 2 / 5 cm | $41.3 \pm 1.0$ | $3.32 \pm 0.21$ | 18.0 | $4.7 \pm 0.3$ |

Table S32. Measured target attainment versus task completion in a separate fixed RoboDojo diagnostic cohort. Attainment requires position within 2 cm, orientation within 15 degrees, and gripper command within 10% of its range at one shared executed step.

| Outcome | Count | Fraction (%) | Diagnostic category |
|---|---|---|---|
| Attained; succeeds | 3521 | 44.0 | Complete target and task |
| Attained; fails | 1585 | 19.8 | Contact or subsequent stage |
| Not attained; succeeds | 351 | 4.4 | Equivalent successful contact |
| Not attained; fails | 2543 | 31.8 | Grounding, steering, or planning |

finement while fixing Sol, the checkpoint, paired seeds, and budgets. The ledger associates each intervention with target error, command/execution discrepancy, contact outcome, and rejection history. Separate pose-term scaling controls examine orientation and gripper balance. Paired contrasts localize the source of a precision change at the task level.

### L.4. Hyperparameter Sensitivity

Table S35 specifies guidance gain, correction cap, active window, pose scales, horizon, executed prefix, temperature multiplier, and confidence threshold. One-factor development sweeps share 20 task instances and five seeds, with the same candidate allowance for competing methods. The default uses gain 1, cap 0.3, window $[0.5, 0.8)$, horizon 50, prefix 16, and threshold $e^{-1}$. The evaluator fixes selected settings before held-out evaluation. Figure S7 displays six sweeps; the accompanying ledger includes pose error, rejection, and cost for each point. Table S36 covers temporal-window and orientation/gripper scales.

### L.5. Failure Taxonomy

Two annotators inspect every failed episode in the diagnostic cohort shared with Table S32, using observations, targets, action projections, controller traces, memory retrieval, and termination status. Each episode receives one primary cause, defined as the earliest identifiable event whose correction could restore task progress, plus optional contributing labels. A third annotator resolves disagreements; both original labels, agreement rate, and adjudication are retained. Ambiguous causal attribution receives an explicit unresolved flag in the detailed ledger. The table uses adjudicated primary labels and all algorithmic failures in this diagnostic cohort as its denominator. Infrastructure failures retain their separate operational category and cost records.

## M. Qualitative Cases and Limitations

### M.1. Complete Interaction Traces

A trace contains the instruction, observation timestamps, target, action chunk, executed prefix, feedback, next decision, and outcome. The following six cases specify executable interaction scenarios and their decision conditions. Each

Table S33. Paired trajectory diagnostics with common observations and initial action noise. Path, acceleration, and jerk use measured translation at the same control rate; saturation and undesired contact are episode percentages. $\Delta A$ is RMS deviation in normalized valid action channels. Mean $\pm$ SD is over five independent runs.

| **Method** | **Path (m)** | **Accel.** $(\mathrm{m/s^2})$ | **Jerk** $(\mathrm{m/s^3})$ | **Sat. (%)** | **Contact (%)** | $\Delta A$ |
|---|---|---|---|---|---|---|
| Raw proposal | $0.67 \pm 0.04$ | $1.36 \pm 0.09$ | $24.3 \pm 0.5$ | 1.8 | 8.9 | $0.00 \pm 0.00$ |
| Post-generation optimization | $0.70 \pm 0.03$ | $2.08 \pm 0.09$ | $34.5 \pm 1.9$ | 4.7 | 9.7 | $0.17 \pm 0.01$ |
| InteractVLA | $0.67 \pm 0.03$ | $1.48 \pm 0.10$ | $26.4 \pm 0.7$ | 2.1 | 7.4 | $0.11 \pm 0.01$ |

Table S34. Intervention sequence for the Precision category. Current-scene targets and corrected controller geometry are diagnostic controls. Each row holds agent, checkpoint, reset seeds, and budget fixed. Mean $\pm$ SD is over five independent runs.

| **Condition** | **SR (%)** | **95% CI** | **Target error (cm)** | **Contact fail. (%)** |
|---|---|---|---|---|
| Language-only | $9.7 \pm 0.6$ | [9.3, 10.2] | $5.16 \pm 0.19$ | 27.0 |
| Online targets | $8.3 \pm 0.7$ | [7.8, 8.8] | $3.78 \pm 0.16$ | 29.0 |
| Current-scene target | $14.8 \pm 0.4$ | [13.6, 16.1] | $0.30 \pm 0.03$ | 22.0 |
| Controller-corrected geometry | $16.2 \pm 1.2$ | [14.6, 17.9] | $0.29 \pm 0.01$ | 19.0 |
| Contact-stage local refinement | $19.7 \pm 2.4$ | [17.3, 22.1] | $0.31 \pm 0.01$ | 14.0 |

Table S35. One-factor development grid for the RoboDojo reference configuration. Every candidate uses the same 20 development task instances and five seeds. The selected defaults are fixed before held-out evaluation.

| **Parameter** | **Candidate values** | **Default** |
|---|---|---|
| Gain | $0.25, 0.5, 1, 2, 4$ | 1 |
| RMS cap | $0.1, 0.2, 0.3, 0.5, 0.8$ | 0.3 |
| Guidance window | $[.3, .6), [.4, .7), [.5, .8), [.6, .9)$ | $[.5, .8)$ |
| Position scale (cm) | $2, 3.5, 5, 7.5, 10$ | 5 |
| Orientation scale (deg) | $10, 20, 30, 45, 60$ | 30 |
| Gripper scale / range | $0.5, 0.75, 1, 1.5, 2$ | 1 |
| Horizon | $16, 32, 50, 64, 80$ | 50 |
| Prefix | $4, 8, 16, 24, 32$ | 16 |
| Temperature multiplier | $0.25, 0.5, 1, 2, 4$ | 1 |
| Confidence threshold | $0, .1, e^{-2}, .2, e^{-1}, .5, .7$ | $e^{-1}$ |

runtime trace adds actual API and tool counts, elapsed time, and sensor or video identifiers.

**Case 1: target redirection.** The raw proposal approaches its habitual interaction location. An agent-specified target changes the generated path toward the intended location. The trace compares both proposals from identical initial noise and records the shared arrival index and residual.

**Case 2: feedback-driven target revision.** The projected approach intersects the wrong image region. The agent uses calibrated correspondence to revise the target, then regenerates from the updated state. The record includes original and revised pixels, depth samples, complete poses, and regenerated chunks.

**Case 3: local trajectory refinement.** A suitable approach has a localized positional discrepancy. A bounded segment edit adds at most 2 cm translation and $10^\circ$ rotation with a smooth endpoint envelope. The trace records segment indices and the before/after commands, followed by measured execution.

**Case 4: rejection and replanning.** A requested target yields $C = e^{-1.4}$, below $e^{-1}$, and returns the rejection record. The revised request yields $C = e^{-0.4}$ and passes the gate. Both residual components, generation indices, state IDs, and consumed computation identify the decision.

**Case 5: unsuccessful recovery.** The contact-failure scenario assumes that the end effector reaches its target and the object slips during transfer. Its trace separates pose attainment from grasp retention, records the attempted recovery, and terminates with the unsatisfied task predicate and its consumed budget.

**Case 6: RSI strategy revision.** In the acceptance-rule example, an offset change has a development completion count of two out of five versus its parent's three out of five, triggering rollback. A subsequent conditional-offset rule has four out of five and preserves protected-skill counts, satisfying acceptance. Strategy versions, validation seeds,

Table S36. Guidance-window and pose-scale sensitivity. Rotation angles specify the equivalent chordal scale; gripper values multiply the full command range. All costs include recovery after rejected or unsuccessful calls. Mean ± SD is over five independent runs.

| Parameter | Value | SR (%) | Error (cm) | Reject (%) | Time (s) |
|---|---|---|---|---|---|
| Window | [.3, .6) | $44.8 \pm 2.7$ | $2.56 \pm 0.21$ | 14.0 | $210 \pm 18$ |
| Window | [.4, .7) | $47.8 \pm 2.1$ | $1.99 \pm 0.16$ | 12.0 | $212 \pm 23$ |
| Window | [.5, .8) | $48.6 \pm 1.9$ | $1.71 \pm 0.15$ | 10.0 | $196 \pm 14$ |
| Window | [.6, .9) | $45.7 \pm 3.0$ | $2.36 \pm 0.14$ | 17.0 | $231 \pm 8$ |
| Rotation scale | 10 deg | $44.9 \pm 2.6$ | $2.58 \pm 0.16$ | 21.0 | $223 \pm 7$ |
| Rotation scale | 20 deg | $48.0 \pm 3.0$ | $1.86 \pm 0.20$ | 13.0 | $204 \pm 20$ |
| Rotation scale | 30 deg | $48.6 \pm 1.9$ | $1.68 \pm 0.16$ | 10.0 | $192 \pm 7$ |
| Rotation scale | 45 deg | $47.9 \pm 2.3$ | $1.81 \pm 0.06$ | 9.0 | $199 \pm 13$ |
| Rotation scale | 60 deg | $44.4 \pm 2.7$ | $2.02 \pm 0.13$ | 8.0 | $197 \pm 10$ |
| Gripper scale | 0.50 range | $45.9 \pm 2.7$ | $2.01 \pm 0.13$ | 19.0 | $220 \pm 11$ |
| Gripper scale | 0.75 range | $46.8 \pm 3.0$ | $1.79 \pm 0.14$ | 13.0 | $194 \pm 12$ |
| Gripper scale | 1.00 range | $48.6 \pm 1.9$ | $1.71 \pm 0.10$ | 10.0 | $201 \pm 17$ |
| Gripper scale | 1.50 range | $47.5 \pm 2.1$ | $1.78 \pm 0.12$ | 8.0 | $201 \pm 9$ |
| Gripper scale | 2.00 range | $44.6 \pm 1.7$ | $2.14 \pm 0.06$ | 7.0 | $215 \pm 8$ |

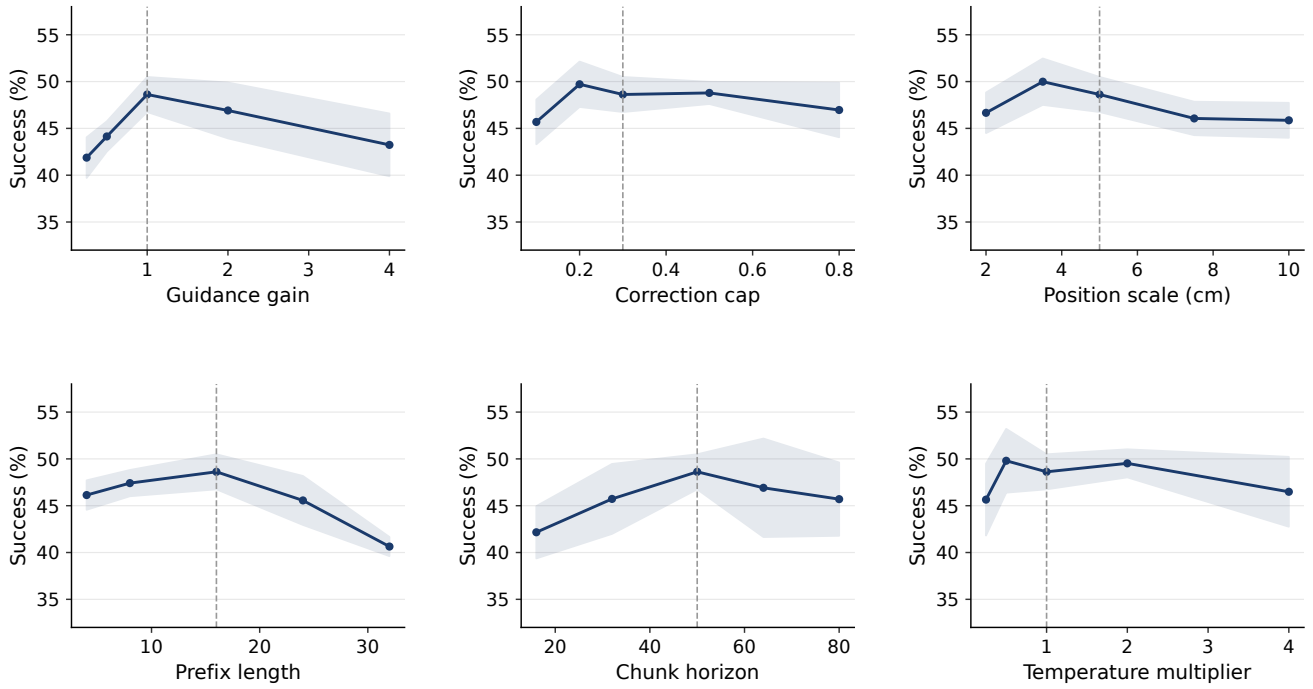



Figure S7. **Sensitivity to guidance and execution settings.** Each panel varies one parameter while retaining the reference configuration for the others. Dashed lines identify the selected settings, and shaded bands indicate one standard deviation across five independent runs. Pose error, rejection, and time use the same paired configurations in the accompanying ledger.

and the separate held-out evaluation records complete the trace specification.

## M.2. Limitations and Scope

The interface assumes differentiable access to the action generator and a command-to-pose mapping consistent with the

Table S37. Primary failure labels for the same separate fixed RoboDojo diagnostic cohort. Percentages use failed algorithmic episodes as their denominator. Infrastructure events are tracked separately from these eight mutually exclusive labels.

| Primary failure | Operational criterion | Count | Fraction (%) |
|---|---|---|---|
| Target grounding | Wrong object or complete-pose target | 765 | 18.5 |
| Task planning | Incorrect stage order or interaction plan | 515 | 12.5 |
| Guidance / mapping | Command mapping or geometric steering error | 606 | 14.7 |
| Contact execution | Slippage, unstable contact, or collision | 1054 | 25.5 |
| Feedback diagnosis | Incorrect interpretation of motion feedback | 357 | 8.6 |
| Rejection / recovery | Rejected-call loop or unproductive recovery | 248 | 6.0 |
| Memory transfer | Transferred rule mismatches the new scene | 256 | 6.2 |
| Budget exhaustion | Task timeout or control/API budget exhausted | 327 | 7.9 |

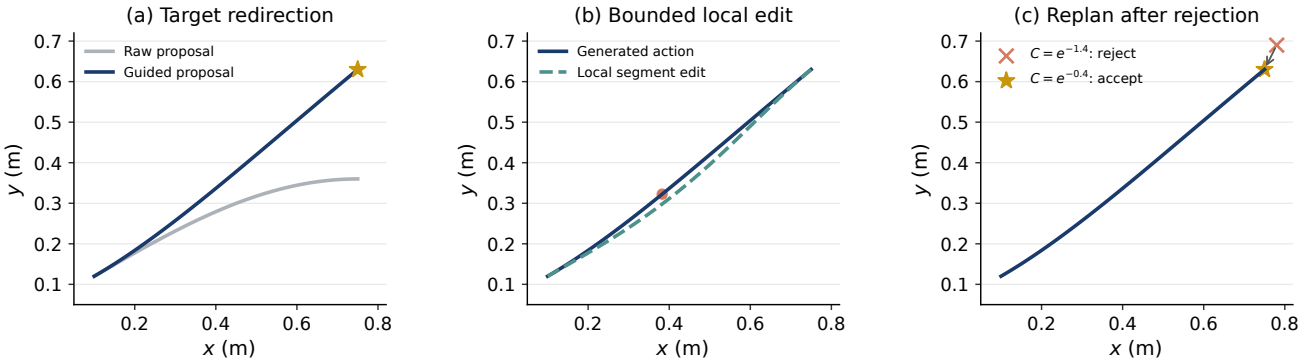



Figure S8. **Geometric interaction traces.** Target redirection changes a raw proposal toward the requested pose; a bounded local edit adjusts a selected motion segment; rejection returns control for target revision. Each panel makes the target, motion, or acceptance decision explicit.

robot controller. Sparse pose targets specify local geometric intent; task sequencing, collision handling, contact forces, and grasp stability require the policy, controller, and agent's complementary capabilities. Soft-time alignment acts within the generated horizon, with subsequent progress determined by execution and replanning.

Reprojection quality depends on calibration, depth, timestamps, and observed execution. Confidence measures target compatibility under the declared residual and correction scales. Semantic grounding and contact feasibility require their own checks. Cross-model transfer, RSI generalization, and efficiency are indexed by the model identifiers, practice distribution, held-out split, and resource budget recorded for each configuration. These dependencies define the conditions for reproducing a result and extending the interface to another platform.

# N. Reproducibility Resources and Evidence Index

## N.1. Implementation and Result Artifacts

The accompanying folder contains the LaTeX sources, table and figure generator, machine-readable data ledgers, assumptions, and artifact-provenance index. From the paper directory, `python3 scripts/build_supplement.py` regenerates the supplementary artifacts, and `python3 scripts/check_supplement.py` checks their schema, denominators, provenance, and document dependencies. Supplying `--compile-log` adds verification of a fresh LaTeX compilation log. The final data-status declaration identifies each artifact's numerical source.

A robot-execution release additionally captures dependency lockfiles; benchmark and checkpoint digests; controller and tool schemas; prompts; adaptation settings; task, seed, and split manifests; accepted and rejected strategy versions; complete trial logs; hardware metadata; and associated videos. The runtime manifest records actual revisions and

provider identifiers at launch. A single-task invocation selects one manifest task and seed with the same configuration used for the complete study. The artifact index links each table row to its originating trial IDs, aggregation recipe, and cost scope.

**Result-record schema.** Each trial records benchmark and task IDs, environment seed, method, agent ID, backbone and checkpoint digest, protocol hash, practice-run ID, strategy version, binary success, normalized score, termination reason, executed steps, API and tool calls, rejection count, and exclusion status. Token categories follow the provider response schema. Timing fields use seconds, translation uses meters, and orientation uses radians internally. A trial links to timestamped observations, targets, proposed actions, executed actions, and immutable memory revisions. The same IDs join outcome, cost, and qualitative records.

## References

[1] Wenrui Bao, Xinxin Liu, Bingxin Xu, and Yuzhang Shang. Recursive video in-context learning for agentic robot. *arXiv preprint arXiv:2610.06843*, 2026. 12, 15

[2] Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Weijie Wan, Baijun Chen, Haoran Lu, Haowen Yan, Honghao Su, et al. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026. 13

[3] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. 12

[4] HexaFuture. HexaShell 1.5: RoboCasa365 submission details. https://github.com/robocasa-benchmark/leaderboard/blob/main/submissions_md/HexaShell-1.5_2026_10_31.md, 2026. Evaluation dated October 31, 2026; accessed November 4, 2026. 13, 15

[5] Shuo Liu, Ishneet Sukhvinder Singh, Yiqing Xu, Jiafei Duan, and Ranjay Krishna. VLS: Steering pretrained robot policies via vision-language models. *arXiv preprint arXiv:2602.03973*, 2026. 15, 17

[6] Manifold AI. $R^2$SI: Trial and error, distilled into capability. https://r2si.worldscape.io/, 2026. 15

[7] PhysicalRSI Team. PhysicalRSI 1.0: Recursive self-harness for scaling embodied skills. https://github.com/PhysicalRSI/PhysicalRSI, 2026. 15

[8] RoboCasa Team. RoboCasa365 leaderboard. https://robocasa.ai/leaderboard, 2026. Snapshot accessed November 4, 2026. 13

[9] RoboDojo Team. RoboDojo simulation leaderboard. https://robodojo-benchmark.com/leaderboard?bench=sim, 2026. Snapshot accessed November 4, 2026; success-rate entries. 15

[10] Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Yi Nie, et al. Harness VLA: Steering frozen VLAs into reliable manipulation primitives via memory-guided agents. *arXiv preprint arXiv:2607.08448*, 2026. 12, 15