CHUNKTRUST: ADAPTING EXECUTION HORIZONS FOR ROBOT POLICIES WITH ACTION-EXPERT EVIDENCE

Anonymous authors
Paper under double-blind review

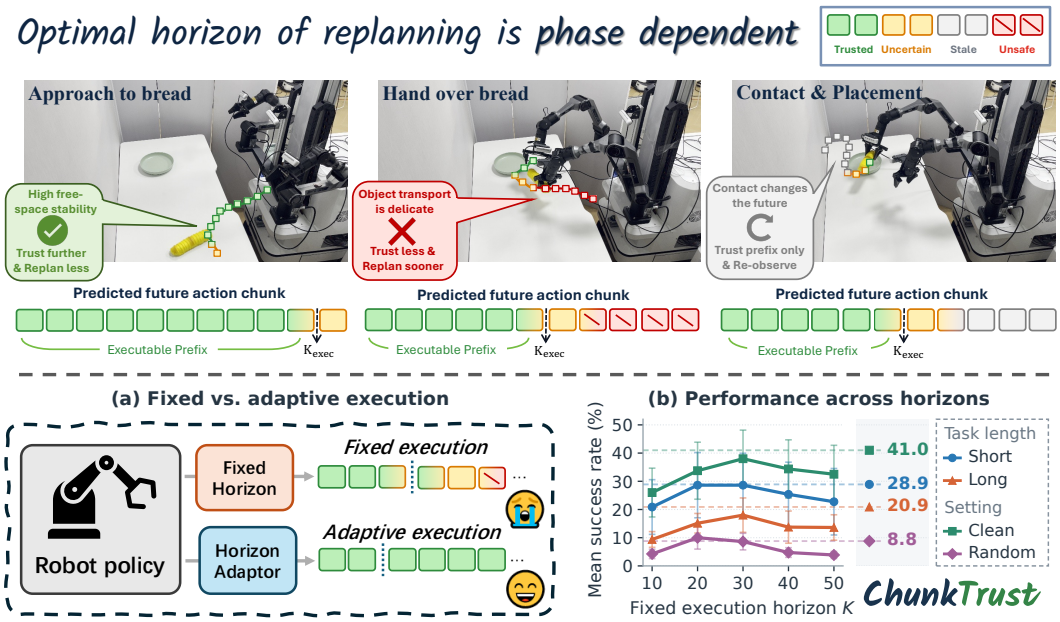

Figure 1: **Fixed execution horizons are brittle across tasks and settings. Top:** the reliable executable prefix is phase dependent in a real bread-manipulation rollout (qualitative trust illustration $K_{\text{exec}} = K_t$). **(a)** A shared robot policy with fixed or adaptive execution horizons. **(b)** Fixed horizons $K \in \{10, 20, 30, 40, 50\}$ versus adaptive execution on π_0 RoboTwin2.0 evaluations, grouped by task length and environment setting. Error bars indicate ± 1 standard error of the mean across task–setting combinations. Dashed lines show the corresponding adaptive references.

ABSTRACT

Robot foundation policies predict action chunks, but how many actions to execute before replanning depends on the current task phase. We introduce ***ChunkTrust***, which treats the execution horizon as a latent variable inferred from action-expert evidence rather than a fixed hyperparameter. Its training-free *Action-aware Horizon Selector* (AHS) combines intra-chunk spectral stability of generation traces with inter-chunk continuity between executed history and predicted actions. An online Beta posterior with kernel forgetting tracks horizon preferences across replans. A lightweight *Query-based Horizon Adapter* (QHA) optionally distills AHS preferences into a dense learned prior, which is fused with fresh online evidence while the base policy remains frozen. Across RoboTwin2.0 and RoboCasa GR1 Tabletop, AHS improves overall task-averaged success for each evaluated base-policy configuration, including gains of $+6.80$ percentage points on $\pi_{0.5}$ over all 50 RoboTwin2.0 tasks and $+9.67$ percentage points on Qwen3GR00T in RoboCasa. AHS+QHA raises the gain over Base to $+9.44$ percentage points on the eight-task $\pi_{0.5}$ evaluation. On four real-world household tasks, AHS improves the equal-task mean normalized process score from 50.4% to 57.5%. Ablations examine the contributions of both evidence terms, temporal memory, and the learned prior.

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

1 INTRODUCTION

Robot foundation policies, including Vision-Language-Action (VLA) models (Black et al., 2024; Intelligence et al., 2025) and World-Action Models (WAMs) (Yuan et al., 2026; Ye et al., 2026b), predict action chunks to amortize inference and maintain temporal coherence. The execution horizon can differ from the generated chunk length, reflecting a trade-off between longer execution that preserves smooth progress but delays correction and shorter execution that increases feedback frequency but can disrupt coherent motion (Lu et al., 2026). In manipulation, this trade-off changes within an episode: free-space approach may tolerate longer open-loop execution, whereas contact or delicate transport demands earlier replanning. Fig. 1 illustrates this phase dependence in a real rollout and shows that the best fixed horizon varies across task lengths and evaluation settings. The resulting *trust boundary* problem is to determine how many actions from the current chunk the robot should execute before replanning.

Existing adaptive chunking methods derive execution horizons from attention structure, action entropy, or cross-horizon agreement (Wang et al., 2026; Liang et al., 2026; Jing et al., 2025). However, an internally consistent prediction can still be incompatible with the motion already executed. Meanwhile, evidence at individual replans can be noisy, while similar execution contexts recur within and across episodes. This raises a central question: *How can a robot identify complementary evidence native to its action expert and internalize it as reusable knowledge for adaptive execution?*

We introduce ChunkTrust, a framework that couples online evidence accumulation with context-conditioned horizon learning (Fig. 3). ChunkTrust evaluates candidate execution prefixes along two complementary dimensions. We assess spectral stability during action generation, drawing on frequency-domain analyses of diffusion and flow models (Si et al., 2024; Huang et al., 2026a). We also assess continuity with recently executed motion, reflecting the importance of cross-chunk consistency in robot control (Liu et al., 2025b; Black et al., 2025). Our retrospective analysis in Fig. 2 provides empirical support for this pairing. Failed episodes exhibit higher median spectral instability and boundary variation, with the highest failure rate observed when both risks are elevated.

ChunkTrust internalizes this evidence through online memory and a learned prior. The *Action-aware Horizon Selector* (AHS) accumulates evidence in an episode-local Beta state with forgetting, retaining useful horizon preferences while adapting to phase changes. The *Query-based Horizon Adapter* (QHA) learns context-conditioned horizon preferences from AHS-derived supervision, enabling their reuse beyond the current episode. At deployment, QHA supplies a dense prior that complements online AHS evidence, combining learned preferences with adaptation to the current rollout.

We evaluate the benefits of online horizon adaptation and learned horizon preferences across RoboTwin2.0, RoboCasa GR1 Tabletop, and four real-world household tasks. AHS improves aggregate success across all evaluated simulation base-policy configurations, including gains of 6.80 percentage points on the complete 50-task RoboTwin2.0 suite and 9.67 percentage points with Qwen3GR00T on RoboCasa. Adding QHA further improves aggregate success on the eight-task evaluation and benefits two tasks excluded from horizon-head training, supporting reuse of the learned preferences beyond the QHA training tasks. On real robots, AHS increases the equal-task mean normalized process score by 7.1 percentage points. Controlled comparisons and ablations examine the contributions of complementary evidence, temporal memory, and the learned prior, alongside alternative horizon selectors and inference costs.

Our contributions are threefold: (1) we formulate execution-horizon adaptation as inference over candidate prefixes, grounded in the action expert's generation dynamics and compatibility with executed history. (2) we introduce AHS, which integrates dual evidence with a phase-aware Beta posterior, and QHA, which learns a context-conditioned horizon prior that complements online evidence without updating the base policy. (3) we evaluate across policy families, two simulation benchmarks, and real robots, with full task-level results and controlled evidence, memory, transfer, and cost analyses.

2 RELATED WORK

Robot policies and action chunking. Mobile ALOHA (Fu et al., 2024b) and Diffusion Policy (Chi et al., 2025) use multi-step action predictions for temporal coherence. Flow-based policies such as

π_0 (Black et al., 2024) and $\pi_{0.5}$ (Intelligence et al., 2025), and world-action models such as Fast-WAM (Yuan et al., 2026), extend action generation to broader task distributions. FASTER (Lu et al., 2026) prioritizes near-term sampling through horizon-aware scheduling and streaming execution. ChainVLA (Huang et al., 2026b) conditions successive queries on task progress and the unexecuted action suffix. ChunkTrust instead selects how much of a chunk to execute before observing again, using the action expert's generation trace without changing its weights or resampling the chunk.

Adaptive horizons and test-time selection. BID (Liu et al., 2025b) selects among sampled chunks using backward coherence and forward contrast, targeting consistency across predictions. Horizon adaptation instead changes the executed prefix: Mixture-of-Horizons uses cross-horizon consensus (Jing et al., 2025), AutoHorizon uses action self-attention as a predictive-limit proxy (Wang et al., 2026), and Adaptive Action Chunking uses action entropy (Liang et al., 2026). HiPolicy combines multi-frequency chunk generation with entropy-guided execution (Zhang et al., 2026). ChunkTrust focuses on the evidence and memory used to select a prefix from a frozen generator. Its two scores measure spectral variation during generation and speed variation after stitching to executed history. An online Beta state accumulates their soft feedback across replans with forgetting. QHA complements this episode-local state with a distilled context-conditioned prior. Appendix G provides broader background.

3 Horizon-Aware Evidence from the Action Expert

3.1 Preliminaries

At replan step t, a frozen policy conditions on visual observation o_t, proprioceptive state $\mathbf{x}_t^{\text{prop}}$, and instruction ℓ. Its backbone produces context tokens $\mathbf{C}_t = f_\theta(o_t, \mathbf{x}_t^{\text{prop}}, \ell)$, and the action expert predicts

$$\hat{\mathbf{A}}_t = [\hat{\mathbf{a}}_{t,1}, \dots, \hat{\mathbf{a}}_{t,H}] \in \mathbb{R}^{H \times d_a}. \tag{1}$$

The controller executes a prefix of length $K_t \in \{1, \dots, H\}$ before re-observation. We score candidates $k \in \mathcal{K} \subseteq \{1, \dots, H\}$ using internal generation stability and compatibility with executed history. The expected-round rule in Sec. 4 can select intermediate integer lengths rather than only grid points.

A single policy call with trace recording returns $(\hat{\mathbf{A}}_t, \mathcal{F}_t) = \pi_\theta(o_t, \mathbf{x}_t^{\text{prop}}, \ell)$. For flow-based action experts (Lipman et al., 2023), $\mathcal{F}_t$ contains velocity predictions recorded during generation, without changing the actions. We write the trace and executed history as

$$\mathcal{F}_t = \{\mathbf{v}_{t,\tau} \in \mathbb{R}^{H \times d_a}\}_{\tau=0}^{T-1}, \qquad \mathcal{H}_t = [\mathbf{a}_{n_t - N_t + 1}^{\text{exec}}, \dots, \mathbf{a}_{n_t}^{\text{exec}}], \tag{2}$$

where τ indexes sampling steps, n_t counts actions executed before replan t, and N_t is the available history length. The following evidence terms use $\mathcal{F}_t$ and $(\mathcal{H}_t, \hat{\mathbf{A}}_t)$, respectively.

3.2 Intra-Chunk Spectral Stability

We measure variation in the velocity-prefix spectrum during generation. Specifically, we apply the Fourier transform along the action horizon at each denoising step and compare the resulting spectra across steps. For each candidate k, we pad its velocity prefix to length H and apply a one-dimensional Fourier transform along the action-horizon axis, separately for action dimensions $j \in \{1, \dots, d_a\}$ and nonnegative frequencies $\omega \in \Omega$:

$$\widehat{\mathbf{V}}_{t,\tau}^{(k)}(\omega, j) = \mathrm{FFT}_h\left(\bar{\mathbf{v}}_{t,\tau}^{(k)}[:, j]\right)_\omega, \quad \text{where} \quad \bar{\mathbf{v}}_{t,\tau}^{(k)} = \mathrm{Pad}_H(\mathbf{v}_{t,\tau,1:k,:}) \in \mathbb{R}^{H \times d_a}, \tag{3}$$

Let $\Omega_{\text{hi}} \subset \Omega$ contain frequencies above cutoff fraction c_{cut} of the discrete frequency grid. The fraction of energy in this band, aggregated across action dimensions, is

$$r_{\text{hi},t,\tau}(k) = \frac{\sum_{\omega \in \Omega_{\text{hi}}} \sum_{j=1}^{d_a} \left|\widehat{\mathbf{V}}_{t,\tau}^{(k)}(\omega, j)\right|^2}{\sum_{\omega \in \Omega} \sum_{j=1}^{d_a} \left|\widehat{\mathbf{V}}_{t,\tau}^{(k)}(\omega, j)\right|^2}. \tag{4}$$

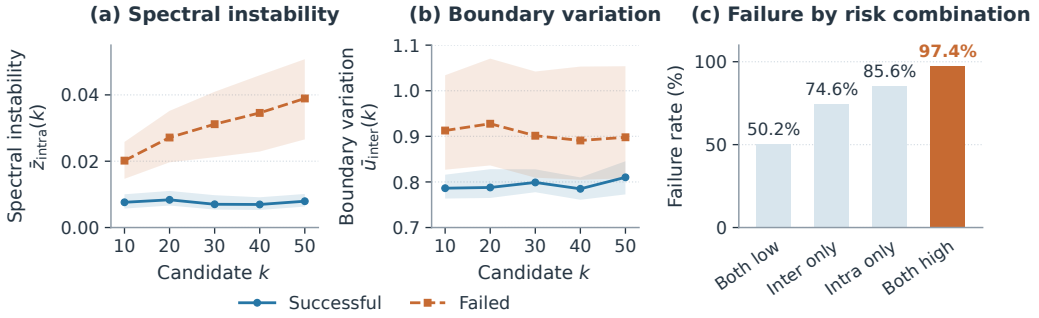

Figure 2: **Horizon-aware evidence and episode outcomes.** Analysis of 1,600 RoboTwin2.0 episodes with π_0 and fixed $K = H = 50$. **(a,b)** Episode means of replan-level evidence: $\bar{z}_{\text{intra}}(k) = R_z^{-1} \sum_t z_{\text{intra},t}(k)$, $\bar{u}_{\text{inter}}(k) = R_u^{-1} \sum_t u_{\text{inter},t}(k)$. Sums and counts R_z, R_u use valid replans for each metric and horizon. Lines/bands show medians/interquartile ranges across episodes. **(c)** Failure rates after median-splitting episode risks (within-episode 75th percentiles of $1 - q_{\text{intra}}$ and $1 - q_{\text{inter}}$). "Inter only"/"Intra only" means only the named risk is high. Details: Appendix B.

High-frequency energy reflects rapid variation along the future-action axis. To track changes during generation, we define an early baseline $\bar{r}_{\text{hi},t,0}(k) = \frac{1}{W_\tau} \sum_{\tau=0}^{W_\tau - 1} r_{\text{hi},t,\tau}(k)$ from the first W_τ sampling steps. The raw intra-chunk instability is its root mean square (RMS) deviation over all denoising steps, including the baseline window:

$$z_{\text{intra},t}(k) = \left[\frac{1}{T} \sum_{\tau=0}^{T-1} \left(r_{\text{hi},t,\tau}(k) - \bar{r}_{\text{hi},t,0}(k) \right)^2 \right]^{1/2}. \tag{5}$$

The early window supplies a reference rather than being discarded from the RMS calculation. The score measures deviation from this reference, not simply the final chunk's high-frequency energy. Because raw scales vary across tasks, policies, and action normalizations, we min-max normalize this score within $\mathcal{K}$ and set $q_{\text{intra},t}(k) = \exp[-\alpha_{\text{intra}} \tilde{z}_{\text{intra},t}(k)]$, where $\alpha_{\text{intra}} > 0$ controls the penalty strength. Larger spectral deviations thus receive lower quality. Normalization makes this a relative comparison among candidate prefixes at the current replan. The resulting quality need not be comparable in absolute scale across unrelated episodes. Fig. 2(a) uses the episode mean $\bar{z}_{\text{intra}}(k)$ of this same replan-level score, with the exact aggregation specified in the caption.

3.3 Inter-Chunk Continuity

An internally stable prefix may still be incompatible with recent motion. For continuity window W_h, we concatenate the available history suffix and candidate future prefix:

$$\mathbf{S}_t^{(k)} = \left[\text{Suffix}_{(W_h - k)_+}(\mathcal{H}_t), \hat{\mathbf{a}}_{t,1}, \ldots, \hat{\mathbf{a}}_{t,k} \right]. \tag{6}$$

Here $(x)_+ = \max(0, x)$. Let $\delta_i^{(k)} = \|\mathbf{S}_{t,i+1}^{(k)} - \mathbf{S}_{t,i}^{(k)}\|_2$ denote first-difference speed along the stitched trajectory. The raw inter-chunk discontinuity is the speed coefficient of variation:

$$u_{\text{inter},t}(k) = \text{Std}\left(\{\delta_i^{(k)}\}_i \right) / \text{Mean}\left(\{\delta_i^{(k)}\}_i \right). \tag{7}$$

This proxy penalizes irregular speed, including boundary jumps and stop-and-go motion. A smaller value indicates a more uniform stitched trajectory, whereas a larger value can reflect an abrupt correction despite a smooth candidate viewed in isolation. As above, we min-max normalize within $\mathcal{K}$ and set $q_{\text{inter},t}(k) = \exp[-\beta_{\text{inter}} \tilde{u}_{\text{inter},t}(k)]$, with $\beta_{\text{inter}} > 0$. Fig. 2(b) reports the corresponding episode mean $\bar{u}_{\text{inter}}(k)$ over valid replans. Failed episodes have higher median raw scores for both evidence terms across the candidate horizons. The shaded bands show interquartile ranges.

3.4 Dual-Evidence Fusion

We combine internal stability and compatibility with recent motion into

$$q_{\text{mix},t}(k) = (1 - \lambda) q_{\text{intra},t}(k) + \lambda q_{\text{inter},t}(k), \qquad \lambda \in [0, 1]. \tag{8}$$

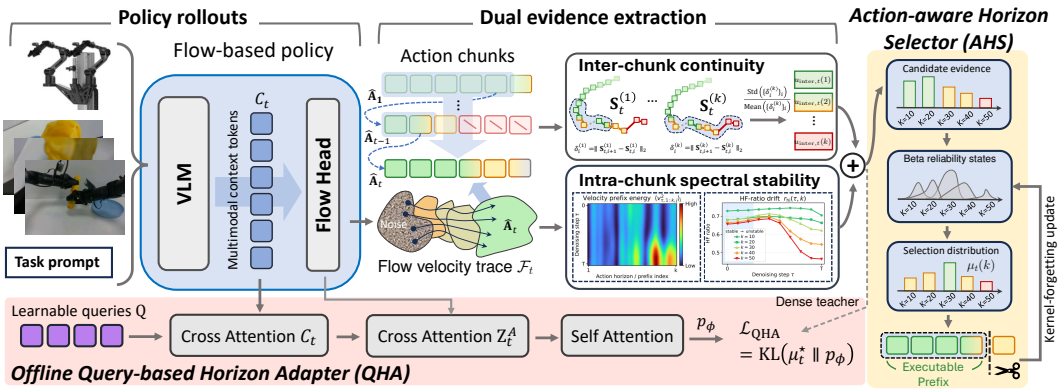

Figure 3: **Overview of horizon adaptation.** AHS normalizes raw evidence, maintains per-candidate Beta reliability, and forms selection distribution μ_t. QHA distills dense evidence into a prior.

The default is $\lambda = 0.5$, with $\lambda = 0$ and $\lambda = 1$ recovering the intra-only and inter-only ablations. This score supplies action-expert evidence for horizon inference, not a deterministic execution rule or a calibrated task-success probability.

Fig. 2(c) summarizes episode risks by the 75th percentiles of $1 - q_{\text{intra}}$ and $1 - q_{\text{inter}}$ over replans, then splits each risk at its median. Failure rises from 50.2% when both risks are low to 97.4% when both are high, with intermediate rates when only one is high. These associations motivate combining the evidence, but they do not establish that a low-risk prefix guarantees success. The diagnostic episodes use fixed $K = H = 50$, so the comparison characterizes the association of evidence with outcomes without selecting trajectories based on AHS decisions. Full distributions and aggregation details are reported in Appendix B.

4 Action-aware Horizon Selection and Query-based Adaptation

AHS uses the candidate quality $q_{\text{mix},t}(k)$ from Sec. 3 to maintain an online reliability posterior. QHA optionally supplies a distilled dense horizon prior (Fig. 3). Neither updates the base action generator.

4.1 Action-aware Horizon Selector

Online reliability posterior. Manipulation alternates between phases such as approach, contact, transport, and placement. A horizon that was useful in a stable phase may become undesirable after a contact change, motivating memory that can also forget. To retain information across noisy replans, AHS maintains a Beta state for each $k \in \mathcal{K}$, initialized by $a_0(k) = b_0(k) = 1$. This is episode-local memory of evidence-derived reliability, not a supervised task-success model. Following Thompson sampling (Russo et al., 2018), we draw a sample and combine it with the current quality:

$$\hat{\xi}_t(k) \sim \text{Beta}(a_t(k), b_t(k)), \qquad s_t(k) = \hat{\xi}_t(k)\, q_{\text{mix},t}(k). \tag{9}$$

Here $q_{\text{mix},t}$ measures the current chunk, while sampled reliability reflects accumulated episode-local evidence. Their product combines both. Except for uniform exploration with probability ϵ_{exp}, temperature scaling and expected-round selection give:

$$\mu_t(k) \propto s_t(k)^{1/T_{\text{sel}}}, \quad \sum_{k\in\mathcal{K}} \mu_t(k) = 1, \qquad K_t = \text{clip}_{[1,H_t^{\text{avail}}]}\, \text{round}\Big[\sum_{k\in\mathcal{K}} k\, \mu_t(k)\Big]. \tag{10}$$

Here H_t^{avail} is the available chunk length. If all scores degenerate, μ_t falls back to uniform over valid candidates. Expected-round combines candidate preferences; exploration samples one valid candidate uniformly. Clipping keeps the prefix within the available prediction. AHS executes $\hat{\mathbf{a}}_{t,1:K_t}$, appends the executed actions to $\mathcal{H}$, and replans from the next observation.

Posterior update with kernel forgetting. After execution, the soft feedback is $y_t = \sum_{k\in\mathcal{K}} \mu_t(k) q_{\text{mix},t}(k)$, or the sampled candidate quality on exploration steps. A Gaussian ker-

Table 1: **AHS performance across simulation benchmarks.** Success rate (%). Each comparison uses the same policy checkpoint. $\Delta = \mathrm{SR}_{\mathrm{AHS}} - \mathrm{SR}_{\mathrm{Base}}$.

Policy	Task Scope	Train Recipe	Base	+AHS	Δ (pp)
RoboTwin2.0	Easy and Hard				
$\pi_{0.5}$	50 tasks	multitask post-training	56.70	**63.50**	+6.80
π_0	8 tasks	task-specific post-training	18.19	**21.38**	+3.19
$\pi_{0.5}$	8 tasks	task-specific post-training	29.63	**36.25**	+6.63
Fast-WAM	8 tasks	multitask post-training	86.81	**88.13**	+1.31
RoboCasa GR1 Tabletop					
$\pi_{0.5}$	24 tasks	multitask post-training	40.08	**42.50**	+2.42
GR00T N1.5	24 tasks	zero-shot	43.25	**44.92**	+1.67
GR00T N1.6	24 tasks	zero-shot	47.61	**51.42**	+3.80
Qwen3GR00T	24 tasks	multitask post-training	47.83	**57.50**	+9.67

nel $w_t(k) \propto \exp[-(k - K_t)^2/(2\sigma_K^2)]$, normalized over $\mathcal{K}$, shares feedback among nearby horizons. Exponential forgetting (Raj & Kalyani, 2017) updates the state:

$$a_{t+1}(k) = \rho_f a_t(k) + \eta\, w_t(k) y_t, \tag{11}$$

$$b_{t+1}(k) = \rho_f b_t(k) + \eta\, w_t(k)(1 - y_t), \tag{12}$$

where ρ_f, η, and σ_K control forgetting, update strength, and neighborhood sharing. Decaying old evidence lets the selector adapt as the task changes phase. The feedback comes from action-expert quality, not an observed task-success label. Neighboring candidates receive shared soft evidence rather than independent rollout outcomes. Kernel sharing couples nearby lengths; forgetting discounts earlier phases.

4.2 Training-Time Query-Based Horizon Adapter

QHA predicts a dense distribution $p_\phi(K_t = h \mid \mathbf{C}_t, \mathbf{Z}_t^A)$ for $h \in \{1, \ldots, H\}$ from VLM context tokens $\mathbf{C}_t$ and action-latent tokens $\mathbf{Z}_t^A$. With a sparse candidate grid, AHS explicitly scores each candidate, while QHA predicts a preference for every action step in one forward pass. Learned horizon queries use bridge cross-attention over both token streams (Fig. 3). Architecture and teacher construction are specified in Appendix A.6. The teacher normalizes dense evidence without online Beta memory:

$$\mu_t^\star(h) \propto q_{\mathrm{mix},t}(h)^{1/T_{\mathrm{teach}}}, \qquad \sum_{h=1}^{H} \mu_t^\star(h) = 1, \qquad h = 1, \ldots, H. \tag{13}$$

Only QHA is trained, minimizing $\mathcal{L}_{\mathrm{QHA}} = \mathrm{KL}(\mu_t^\star \,\|\, p_\phi)$. QHA thus encodes horizon preferences across training episodes in its parameters, providing cross-episode memory that complements AHS's episode-local state. At deployment, let $\tilde{q}_{\mathrm{mix},t}(h)$ denote quality on the dense grid, interpolated from sparse evidence or scored directly on that grid. QHA supplies a prior, combined with dense Beta reliability samples as

$$s_{\mathrm{eff},t}(h) = \hat{\xi}_t(h)\, \tilde{q}_{\mathrm{mix},t}(h)\, p_\phi(K_t = h \mid \mathbf{C}_t, \mathbf{Z}_t^A)^\gamma, \qquad h = 1, \ldots, H, \tag{14}$$

where $\gamma \geq 0$ controls prior strength. Applying temperature scaling and the selection rule above to this dense score yields AHS+QHA, combining learned preferences with current episode evidence. The dense prior supplies a context-conditioned preference before the episode-local state has accumulated much evidence. It does not remove the need to record generation traces or evaluate AHS candidates in the hybrid setting, as those computations provide the online correction to the prior.

5 Experiments

We evaluate on RoboTwin2.0 (Chen et al., 2025), RoboCasa GR1 Tabletop (Nasiriany et al., 2024), and real robots. Each Base–AHS comparison fixes the policy checkpoint. AHS changes execution

Table 2: **QHA augmentation and transfer on RoboTwin2.0.** Success rate (%). Easy and Hard denote clean and randomized settings. **A:** Evaluation on the eight tasks used to train QHA. **B:** For $\pi_{0.5}$, QHA is trained on six tasks and evaluated on two held-out tasks. Results are averaged over Easy and Hard. Bold marks the best method for each policy and evaluation condition.

A. QHA training-task evaluation

Task	π_0 (Black et al., 2024)						$\pi_{0.5}$ (Intelligence et al., 2025)					
	Base		AHS		AHS+QHA		Base		AHS		AHS+QHA	
	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Blocks Ranking RGB	19.00	0.00	28.00	1.00	**31.00**	**5.00**	36.00	20.00	48.00	29.00	**56.00**	**33.00**
Handover Block	41.00	**10.00**	41.00	5.00	**64.00**	9.00	**44.00**	**14.00**	**44.00**	**14.00**	32.00	10.00
Handover Mic	**100.00**	2.00	**100.00**	24.00	**100.00**	**30.00**	98.00	**64.00**	**100.00**	56.00	98.00	59.00
Hanging Mug	17.00	3.00	14.00	9.00	**23.00**	**10.00**	**14.00**	8.00	13.00	12.00	13.00	**14.00**
Place A2B Left	24.00	**1.00**	34.00	0.00	**39.00**	**1.00**	41.00	4.00	**47.00**	4.00	**47.00**	**15.00**
Place Bread Basket	11.00	8.00	**18.00**	**16.00**	15.00	10.00	30.00	17.00	50.00	29.00	**55.00**	**41.00**
Place Bread Skillet	15.00	2.00	**23.00**	**5.00**	**23.00**	2.00	24.00	10.00	36.00	**19.00**	**44.00**	**19.00**
Place Can Basket	**33.00**	**5.00**	22.00	2.00	**33.00**	3.00	35.00	15.00	50.00	29.00	**55.00**	**34.00**
Mean by setting	32.50	3.88	35.00	7.75	**41.00**	**8.75**	40.25	19.00	48.50	24.00	**50.00**	**28.13**
Overall	18.19		21.38		**24.88**		29.63		36.25		**39.06**	

B. QHA transfer to held-out tasks ($\pi_{0.5}$)

Task	Base	AHS	AHS+QHA	Δ vs. AHS (pp)
Blocks Ranking RGB	28.00	38.50	**43.50**	+5.00
Place Bread Basket	23.50	41.50	**42.00**	+0.50
Overall	25.75	40.00	**42.75**	+2.75

without updating policy weights. Simulation tables report success rates (%). In Tabs. 1 and 2, tasks are weighted equally after averaging Easy and Hard within each RoboTwin2.0 task. Rounding follows aggregation. Easy and Hard denote clean and randomized settings. Appendix A specifies checkpoints, candidate horizons, and evaluation protocols.

5.1 AHS ACROSS SIMULATION BENCHMARKS

Tab. 1 tests AHS across benchmarks and training regimes. Its task scope and training recipe distinguish evaluation coverage from checkpoint provenance. Absolute scores across regimes are not matched-policy comparisons. The evaluation spans task-specific post-training, multitask post-training, and zero-shot deployment on the target benchmark. Gains in these regimes test whether execution adaptation remains useful across the evaluated backbones. They do not imply that AHS repairs every failed task or replaces policy training.

RoboTwin2.0. The full-suite evaluation uses one multitask-post-trained $\pi_{0.5}$ (Intelligence et al., 2025) checkpoint on 50 tasks, with 20 rollouts per task and setting. AHS improves success from 56.70% to 63.50% (+6.80 percentage points). The eight-task evaluation uses task-specific π_0 (Black et al., 2024) and $\pi_{0.5}$ checkpoints and multitask Fast-WAM (Yuan et al., 2026), with 100 rollouts per task and setting. All three improve in aggregate. These are distinct checkpoint and evaluation cohorts, not subset and full-suite results for one policy. Fast-WAM improves from 86.81% to 88.13%, showing that execution adaptation can still help a stronger baseline, although its aggregate gain is smaller than those of the task-specific policies. Appendix Tabs. 8 and 9 retain full per-task outcomes, including regressions.

RoboCasa GR1 Tabletop. We evaluate all 24 tasks; the $\pi_{0.5}$ control uses 50 rollouts per task. The multitask-post-trained $\pi_{0.5}$ and Qwen3GR00T (Ye et al., 2026a; Community, 2026) improve from 40.08% to 42.50% and from 47.83% to 57.50%, respectively. Both zero-shot Isaac-GR00T baselines (Bjorck et al., 2025) also improve in aggregate. All four use $\mathcal{K} = \{4, 8, 12, 16\}$. Appendix

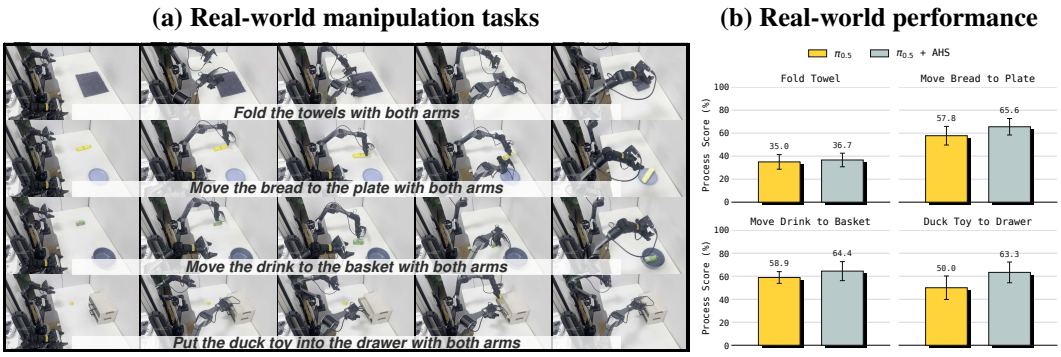

Figure 4: **Real-world deployment. (a)** Four bimanual household tasks: fold towels, move bread to a plate, move a drink to a basket, and put a duck toy into a drawer. **(b)** Normalized process scores for $\pi_{0.5}$ and $\pi_{0.5}$ + AHS over 15 rollouts per task.

Table 3: **Candidate-set scaling on $\pi_{0.5}$.** Success and per-replan overhead over four RoboTwin2.0 tasks as $|\mathcal{K}|$ increases. Fixed denotes $\pi_{0.5}$ without horizon selection. ΔSuccess is in percentage points.

Candidate set	$\lvert\mathcal{K}\rvert$	Success (%)	ΔSuccess	Replan time (ms)	Overhead (%)	Replan counts	Ep. time (s)
Fixed	–	22.00	–	95.83	–	4.22	22.43
$\mathcal{K}_5$	5	33.00	+11.00	96.50	0.70	8.36	23.17
$\mathcal{K}_{10}$	10	32.50	+10.50	96.84	1.05	9.34	23.08
$\mathcal{K}_{50}$	50	**34.88**	**+12.88**	99.28	3.60	10.57	**21.39**

Tab. 10 provides the complete breakdown. Fixed-horizon, action-only, temporal-memory, and AAC (Liang et al., 2026) comparisons appear in Appendix C.5.

5.2 QHA Augmentation and Transfer

Augmentation on QHA training tasks. Each policy family uses one QHA head trained on eight tasks with the base policies frozen (Tab. 2A). AHS+QHA improves overall success over AHS from 21.38% to 24.88% for π_0 and from 36.25% to 39.06% for $\pi_{0.5}$. Both Easy and Hard means improve, but $\pi_{0.5}$ regresses on Handover Block in both settings. This may reflect a mismatch between the shared horizon prior and the feedback timing required during bimanual object transfer.

Transfer to tasks held out from QHA training. A separate $\pi_{0.5}$ head trains on six tasks and tests on two held-out tasks (Tab. 2B). AHS+QHA improves the equal-task mean from 40.00% to 42.75%. Blocks Ranking RGB improves from 38.50% to 43.50%, compared with 41.50% to 42.00% on Place Bread Basket. Only the horizon head is task-held-out: base policies remain task-specific. Appendix C.1 gives the split, per-setting results, QHA-only control, and decision agreement.

5.3 Real-World Deployment

We deploy $\pi_{0.5}$ on the AgileX COBOT Magic ALOHA-style bimanual platform, comparing fixed $K = 25$ with AHS over 15 rollouts on each of four household tasks (Fig. 4). Object positions vary across rollouts to test spatial generalization. Scores average predefined sub-steps and rollouts to measure partial progress, rather than binary success (Appendix A.1). AHS raises the equal-task mean from 50.4% to 57.5%, with the largest gain on Duck Toy to Drawer: 50.0% to 63.3%.

5.4 Ablation Studies

All three studies use Place A2B Left, Place Bread Basket, Place Bread Skillet, and Place Can Basket, with 100 rollouts per task and condition. All three studies include Easy and Hard.

Candidate-set scaling. On $\pi_{0.5}$, five candidates raise success from 22.0% with fixed $K = 50$ to 33.0%, versus 34.9% with 50 candidates (Tab. 3). Added per-replan cost rises from 0.67 to 3.45 ms, or 0.70–3.60% of policy inference time. Sparse candidates thus capture most of the gain. The table's episode times cover successes only, so they do not establish all-attempt cost. Appendix D.1 reports all-episode costs for additional cohorts. Appendix E gives hyperparameter sweeps.

Component ablation. On π_0, combining both evidence terms without memory gives 14.6% overall but 3.75% on Hard (Tab. 4). Full AHS reaches 15.0% overall and the best Hard result, 5.75%, supporting temporal memory. QHA raises overall success to 15.8% while Hard falls to 4.00%. Appendix C.5 compares memory designs with shared evidence and candidates.

Table 4: **Component ablation (π_0).** Four-task mean success. Top two: bold/underline.

Method	Easy (%)	Hard (%)	Overall (%)
Base π_0 (default)	20.75	4.00	12.38
Inter-chunk only	23.50	4.00	13.75
Intra-chunk only	23.75	<u>5.25</u>	14.50
Both evidence terms, no posterior	<u>25.50</u>	3.75	14.63
Full AHS	24.25	**5.75**	<u>15.00</u>
Full AHS+QHA	**27.50**	4.00	**15.75**

Latency and asynchronous execution. We combine AHS with real-time chunking (RTC; Black et al., 2025) on task-specific $\pi_{0.5}$ policies (Fig. 5). All predict $H = 50$ actions. Fixed uses target budget $K = 40$, and AHS scores candidates in $\{10, 20, 30, 40\}$ without QHA. At +200 ms, RTC reduces AHS waiting from 7.893 to 0.344 s/episode on Hard and from 6.310 to 0.344 s on Easy. Within RTC, AHS lowers mean observation age from 4.92 to 3.07 s on Hard and from 5.00 to 3.18 s on Easy, while SR rises from 22.00% to 26.75% and from 39.00% to 42.75%, respectively. This requires more policy calls and model computation (Appendix D.2). RTC does not uniformly improve SR over synchronous AHS: at +200 ms on Easy, SR is 42.75% versus 44.75%.

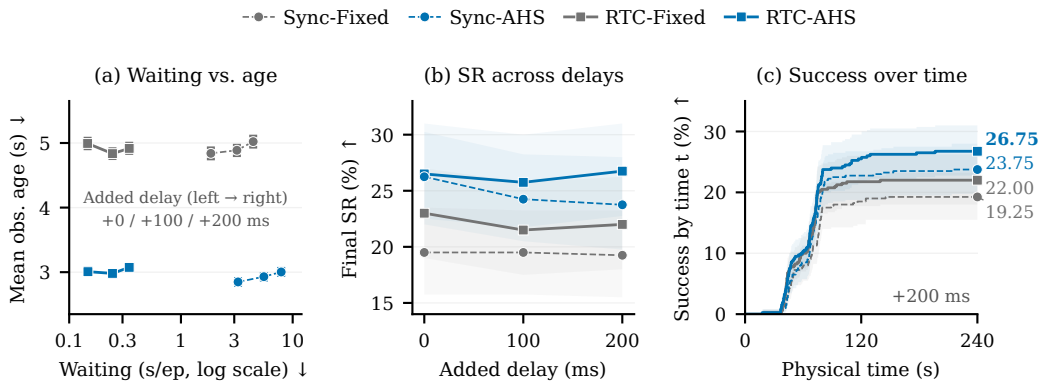

Figure 5: **AHS with asynchronous execution on $\pi_{0.5}$ Hard.** (a) Mean waiting and observation age. Points run left to right as +0/+100/+200 ms additional delay. (b) Final SR. (c) SR(t) at +200 ms. All 400 outcomes per condition are included. Bars and shading are marginal/pointwise 95% paired-seed bootstrap intervals within four tasks. Time uses a controlled physical clock with a 142 ms base delay, not native deployment timing. Easy curves and full results: Appendix D.2.

6 Conclusion

ChunkTrust adapts robot-policy execution horizons using action-expert evidence. AHS combines spectral stability, inter-chunk continuity, and online temporal memory, while optional QHA training supplies a distilled horizon prior without changing the base policy. Experiments support aggregate gains across two simulation benchmarks and real-world manipulation, while transfer, ablation, and cost analyses characterize where adaptation helps. The method requires accessible generation traces, and gains are not uniform across tasks. The evaluation also separates per-replan overhead from episode-level cost: a small selector cost does not imply identical total rollout time. Task breakdowns likewise expose regressions that aggregate gains can conceal, particularly when a learned prior is combined with online evidence. Future work includes multi-chunk horizon inference, cross-policy transfer of the learned prior, and broader real-world validation across hardware and domain shifts.

DISCLOSURE OF AI USE

We used generative AI tools to assist with experimental design, code implementation and debugging, analysis and interpretation of experimental results, figure and table preparation, and manuscript drafting and editing, including clarification of mathematical notation and methodological descriptions. The authors reviewed the AI-assisted code, analyses, figures, and text and take responsibility for the final content of this work, including all claims and artifacts.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

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

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

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 33383–33407. Curran Associates, Inc., 2025. doi: 10.52202/085713-1122. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/300ccb2187dedd4edcc07f7e76d8e553-Paper-Conference.pdf.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S Ryoo, Grecia Salazar, Pannag R Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics Transformer for Real-World Control at Scale. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.025.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

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

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

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-e: An embodied multimodal language model. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara

Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8469–8488. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/driess23a.html.

Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation using low-cost whole-body teleoperation. In *8th Annual Conference on Robot Learning*, 2024a.

Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile ALOHA: Learning bimanual mobile manipulation using low-cost whole-body teleoperation. In *8th Annual Conference on Robot Learning*, 2024b. URL https://openreview.net/forum?id=FO6tePGRZj.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.090.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Guanbo Huang, Jingjia Mao, Fanding Huang, Fengkai Liu, Xiangyang Luo, Yaoyuan Liang, Jiasheng Lu, Xiaoe Wang, Pei Liu, Ruiliu Fu, Ruqi Huang, and Shao-Lun Huang. Exposure bias can alleviate itself via directional and frequency rectification in flow matching. In *European Conference on Computer Vision (ECCV)*, 2026a. URL https://eccv.ecva.net/virtual/2026/poster/5433.

Yuzhi Huang, Weijue Bu, Ziyi Xiong, Jie Wu, Fanding Huang, Jingyan Jiang, and Zhi Wang. ChainVLA: Chaining vision-language-action queries through a unified execution state for long-horizon manipulation. *arXiv preprint arXiv:2608.02326*, 2026b.

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

Dong Jing, Gang Wang, Jiaqi Liu, Weiliang Tang, Zelong Sun, Yunchao Yao, Zhenyu Wei, Yunhui Liu, Zhiwu Lu, and Mingyu Ding. Mixture of horizons in action chunking. *arXiv preprint arXiv:2511.19433*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=ZMnD6QZAE6.

Yuanchang Liang, Xiaobo Wang, Kai Wang, Shuo Wang, Xiaojiang Peng, Haoyu Chen, David Kim Huat Chua, and Prahlad Vadakkepat. Adaptive action chunking at inference-time for vision-language-action models. *arXiv preprint arXiv:2604.04161*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1b: a diffusion foundation model for bimanual manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=yAzN4tz7oI.

Yuejiang Liu, Jubayer Ibn Hamid, Annie Xie, Yoonho Lee, Max Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=qZmn2hkuzw.

Yuxiang Lu, Zhe Liu, Xianzhe Fan, Zhenya Yang, Jinghua Hou, Junyi Li, Kaixin Ding, and Hengshuang Zhao. FASTER: Rethinking real-time flow VLAs. *arXiv preprint arXiv:2603.19199*, 2026.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *RSS 2024 Workshop: Data Generation for Robotics*, 2024.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Vishnu Raj and Sheetal Kalyani. Taming non-stationary bandits: A Bayesian approach. arXiv preprint arXiv:1707.09727, 2017. URL https://arxiv.org/abs/1707.09727.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018.

Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. FreeU: Free lunch in diffusion U-Net. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4733–4743, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Si_FreeU_Free_Lunch_in_Diffusion_U-Net_CVPR_2024_paper.html.

Junhyuk So, Chiwoong Lee, Shinyoung Lee, Jungseul Ok, and Eunhyeok Park. Improving generative behavior cloning via self-guidance and adaptive chunking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=GctsZXLCpl.

BAAI RoboBrain Team, Mingyu Cao, Huajie Tan, Yuheng Ji, Xiansheng Chen, Minglan Lin, Zhiyu Li, Zhou Cao, Pengwei Wang, Enshen Zhou, et al. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Haoxuan Wang, Gengyu Zhang, Yan Yan, Ramana Rao Kompella, and Gaowen Liu. Vla knows its limits. *arXiv preprint arXiv:2602.21445*, 2026.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. In *9th Annual Conference on Robot Learning*, 2025.

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. Starvla-α: Reducing complexity in vision-language-action systems. *arXiv preprint arXiv:2604.11757*, 2026a.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Naveed Malik, Kyungmin Lee, William Liang, Nadun Ranawaka Arachchige, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Danfei Xu, Yilun Du, Ryan Julian, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World action models are zero-shot policies. In *ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling*, 2026b. URL https://openreview.net/forum?id=cd33uUB609.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Jiyao Zhang, Zimu Han, Junhan Wang, Xionghao Wu, Shihong Lin, Jinzhou Li, Hongwei Fan, Ruihai Wu, Dongjiang Li, and Hao Dong. Hipolicy: Hierarchical multi-frequency action chunking for policy learning. *arXiv preprint arXiv:2604.06067*, 2026.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Tai Wang, Ya-Qin Zhang, Jingjing Liu, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kt51kZH4aG.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

ChunkTrust: Adapting Execution Horizons for Robot Policies with Action-Expert Evidence Appendix

A Experimental Setup

We specify evaluation cohorts, checkpoints, scoring criteria, and implementation settings for the results in Sec. 5.

A.1 Real-world Experiments

Hardware Setup. We conduct real-world experiments on an AgileX COBOT Magic platform configured as an ALOHA-style bimanual system (Fu et al., 2024a;b), as shown in Fig. 6. The platform consists of four 6-DoF Piper arms, with two leader arms used for human teleoperation and two follower arms used for data collection and autonomous policy execution. The perception system includes three RealSense D435 cameras: one front-view camera and two wrist-mounted cameras, one on each follower arm.

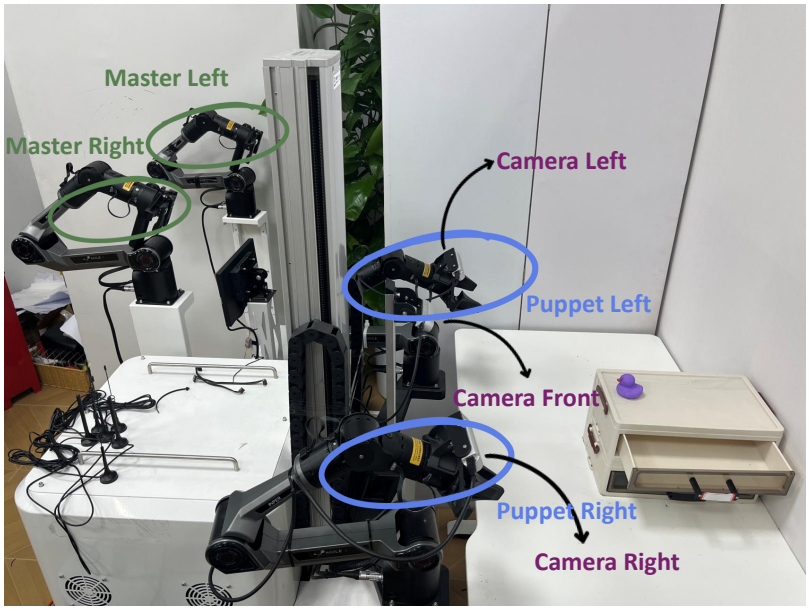

Figure 6: **Real-world robot platform.** AgileX COBOT Magic configured as an ALOHA-style bimanual system for teleoperation, data collection, and autonomous policy execution.

Tasks and evaluation. We collect 200 human-teleoperated demonstrations for each of four bimanual tasks and evaluate $\pi_{0.5}$ with fixed $K = 25$ or AHS over 15 rollouts per task. Data are recorded at 30 FPS, with task-relevant object positions varied across evaluation rollouts. Instructions are "Fold the towel with both arms," "Move the bread to the plate with both arms," "Move the drink to the basket with both arms," and "Put the duck toy into the drawer with both arms." These tasks cover approach, contact, transport, alignment, and placement (Fig. 4).

Process scores. Each sub-step receives 0 for failure, 0.5 for recovered or imperfect completion, and 1 for smooth, accurate completion. We average equally over the task's sub-steps and its rollouts, reporting the result as a percentage. Table 5 preserves the task-specific criteria. In the drawer task, the arm assignment depends on the layout: one arm opens/closes the drawer and the other manipulates the toy.

Table 5: **Real-world scoring criteria.** Scoring is independent of arm assignment.

Task and sub-step	0	0.5	1
Bread: grasp	Fails to grasp	Multiple attempts	Smooth first attempt
Bread: handover	Handover fails	Unstable/awkward receiving grasp	Stable, aligned grasp
Bread: place on plate	Not placed	Poor alignment or rough placement	Clean placement
Drink: push	Only tilts, no useful displacement	Acceptable position, tilt/misalignment	Good position, stable alignment
Drink: grasp	Fails to grasp	Multiple attempts	Smooth first attempt
Drink: place in basket	Fails or drops drink	Rough placement/collision	Clean placement
Towel: first fold, second fold	Not folded over	Folded, misaligned	Folded, well aligned
Duck: open drawer, grasp toy, place toy, close drawer	Sub-step fails	Multiple attempts	Smooth first attempt

A.2 Simulation Experiments

RoboTwin2.0 (Chen et al., 2025) contains 50 bimanual manipulation tasks with strong domain randomization. The complete-suite evaluation in Tab. 1 uses one multitask $\pi_{0.5}$ checkpoint on all 50 tasks, with 20 rollouts per task, setting, and method (2,000 per method). The eight-task evaluation uses 100 rollouts per task, setting, and method (1,600 per method), on Blocks Ranking RGB, Handover Block, Handover Mic, Hanging Mug, Place A2B Left, Place Bread Basket, Place Bread Skillet, and Place Can Basket. Each task is evaluated under two settings: *Easy* (clean scenes) and *Hard* (randomized object poses, lighting, and distractor placement). Results report success rate, averaged equally across the tasks and settings in each cohort.

RoboCasa GR1 Tabletop (Nasiriany et al., 2024) provides 24 pick and-place tasks spanning everyday object categories and novel source–target combinations. We evaluate all 24 tasks, using 50 rollouts per task for the $\pi_{0.5}$ control. Historical policy results retain their reported evaluation budgets and precision. Task success is determined by the native RoboCasa success checker.

A.3 Base Policy Checkpoints

For the eight-task RoboTwin2.0 evaluation, checkpoints follow the protocol of each base policy:

- π_0 (Black et al., 2024): we train LoRA adapters (Hu et al., 2022) with the released RoboTwin2.0 fine-tuning protocol. For each evaluated task, one adapter is trained on that task's clean50 demonstrations and then evaluated. The action chunk size is $H = 50$, with a flow-matching action expert (Lipman et al., 2023) using 10 sampling steps.
- $\pi_{0.5}$ (Intelligence et al., 2025): we use the official RoboTwin2.0 fine-tuning recipe for full-parameter fine-tuning. For each evaluated task, one task-specific checkpoint is trained on clean50 demonstrations and then evaluated. The action chunk size is $H = 50$, with a flow-matching action expert (Lipman et al., 2023).
- X-VLA (Zheng et al., 2026): we evaluate the released X-VLA RoboTwin2.0 checkpoint. The action chunk size is $H = 30$, with a flow-matching action expert.
- Fast-WAM (Yuan et al., 2026): we evaluate the released Fast-WAM multitask RoboTwin2.0 checkpoint on our eight-task subset, rather than post-training a separate policy for each task. The action chunk size is $H = 32$, with a flow-matching action expert.

Multitask $\pi_{0.5}$ checkpoints. The complete-suite controls initialize from the official `pi05_base` and use benchmark-specific full-parameter post-training. RoboTwin2.0 uses 50 clean demonstrations per task (2,500 trajectories and 549,787 frames). RoboCasa GR1 uses 24,000 demonstrations and 6,020,058 frames across 24 tasks. Both use task-uniform sampling, one global quantile normalizer

per benchmark, global batch size 256, seed 42, and eight H100 GPUs for training, with the checkpoint fixed at step 30,000. RoboTwin uses $H = 50$ and Base $K = 50$, while RoboCasa uses $H = 16$ and Base $K = 16$. The latter maps the raw 44-dimensional GR1 interface to 29 effective absolute controls in the padded policy interface. These controls share a policy architecture, not weights across benchmarks. Within each benchmark, Base and AHS use identical policy weights. Their evaluation runs on one RTX 4090.

For the other RoboCasa policies, Isaac-GR00T N1.5 and Isaac-GR00T N1.6 (Bjorck et al., 2025) use released base checkpoints without additional benchmark post-training. Qwen3GR00T uses the released 24-task multitask-post-trained GR1 checkpoint from StarVLA (Ye et al., 2026a; Community, 2026). Thus, "zero-shot" in Tab. 1 describes benchmark adaptation, not an absence of robot pretraining.

A.4 AHS Hyperparameter Configurations

Candidate sets and continuity windows depend on the policy and benchmark:

Evaluation	Candidate horizons $\mathcal{K}$	Continuity window W_h
RoboTwin2.0, π_0 and $\pi_{0.5}$	$\{10, 20, 30, 40, 50\}$	60
RoboTwin2.0, X-VLA and Fast-WAM	$\{10, 20, 30\}$	40
RoboCasa GR1, all policies	$\{4, 8, 12, 16\}$	20

The scaling ablation compares $\mathcal{K}_5 = \{10, 20, 30, 40, 50\}$, $\mathcal{K}_{10} = \{5, 10, \ldots, 50\}$, and $\mathcal{K}_{50} = \{1, \ldots, 50\}$. Shared defaults are $\alpha_{\text{intra}} = \beta_{\text{inter}} = 4$, $\lambda = 0.5$, $a_0 = b_0 = 1$, $\rho_f = 0.99$, $\eta = 1$, and $\sigma_K = 10$. The original eight-task study uses expected-round selection at $T_{\text{sel}} = 0.8$, with Thompson exploration probability $\epsilon_{\text{exp}} = 0.05$. The full50 evaluation uses $T_{\text{sel}} = 1.0$, and the RoboCasa $\pi_{0.5}$ control uses 0.8.

Spectral scoring zero-pads each velocity prefix to H before FFT, uses the first $W_\tau = 3$ denoising steps as its reference, window RMS for z, and cutoff fraction $c_{\text{cut}} = 0.25$. Without sufficient executed history, AHS uses intra-chunk-only scoring ($\lambda = 0$). Costs appear in Appendix D.1.

A.5 Evaluation Protocol

For every (base policy, task, setting) combination, we run a fixed number of rollouts with AHS enabled. Each rollout starts from the standard initial state distribution of the benchmark. The online AHS posterior is reset to its prior ($a_0 = b_0 = 1$) at the beginning of each episode, so its rollout history is episode-local. When QHA is used, its learned weights are reused across episodes without online training. Success is determined by the benchmark's native success checker. We report the raw success rate (percentage of successful rollouts) and equal-weight averages over the stated task and setting groups. Absolute differences are in percentage points. The Δ (pp) columns subtract the corresponding success-rate percentages, rather than reporting relative percentage changes.

In the 50-task evaluation, 62 of the 100 task–setting conditions use exact same-seed and instruction pairing between Base and AHS. The other 38 use deterministic, method-specific fallback identities selected without outcome information after repeated scene-construction failures. The full-suite averages include both groups. Identical episode identities are not assumed for the fallback group.

For the main RoboTwin2.0 benchmarks, Fast-WAM inference uses one NVIDIA H100 (80 GB); π_0, $\pi_{0.5}$, and X-VLA use one RTX 4090 (24 GB). The RoboCasa $\pi_{0.5}$ control also uses one RTX 4090. Hardware metadata for separate runtime profiles is discussed in Appendix D.1. Replanning frequency depends on the selected horizon and execution scheduler.

A.6 QHA Training Configurations

Inputs and teacher. For Tab. 2A, we train one shared QHA per policy family (π_0 or $\pi_{0.5}$) across all eight tasks, keeping the task-specific base generators frozen. Training uses their clean50 Aloha-AgileX LeRobot datasets, with task-balanced batches of 256 (32 per task). Inputs comprise three RGB streams, robot state, action windows, VLM context tokens $\mathbf{C}_t$ with mask $\mathbf{M}_t$, and action-latent tokens $\mathbf{Z}_t^A$ from the final chunk. The action-query window includes 64 history and 50 future steps.

The frozen checkpoint generates the chunk $\hat{\mathbf{A}}_t$, evidence $\mathcal{F}_t$, and dense teacher $\mu_t^\star$ online. The teacher follows Eq. 13 over $h = 1, \ldots, 50$ at $T_{\text{teach}} = 1$, using current evidence without episode-local Beta memory. The six-task held-out protocol is specified separately in Appendix C.1.

Architecture and optimization. Context and action tokens are projected into a shared $d = 256$ space with positional encodings. One bridge layer updates eight learned queries by cross-attending to masked context, then action tokens, followed by self-attention, each with residual connections (Vaswani et al., 2017). It uses eight attention heads, zero dropout, and an enabled fusion gate. Query pooling and an MLP produce 50 logits followed by a softmax. We minimize $\mathrm{KL}(\mu_t^\star \| p_\phi)$ with weight 1 and numerical floor 10^{-6}. Only QHA parameters are updated and the base flow-matching loss is zero. AdamW runs for 10,000 steps with batch size 256, global-norm clipping 1, weight decay 10^{-10}, and EMA 0.99. The warmup-cosine schedule uses 1,000 warmup steps, peak learning rate 5×10^{-5}, and final rate 5×10^{-6}.

Deployment. QHA provides the dense prior in Eq. 14. The sparse-evidence variant interpolates $q_{\text{mix},t}(k)$ from $\mathcal{K}$ onto $\{1, \ldots, H\}$ before Beta sampling. The dense-evidence variant scores each horizon directly, as in the held-out study. Both maintain dense Beta states and apply temperature scaling, expected-round selection, and uniform exploration as in Sec. 4.1. Posterior feedback uses prior-weighted quality $\tilde{q}_{\text{mix},t}(h) p_\phi(h)^\gamma$. The held-out study sets $\gamma = 1$.

B Action-Expert Evidence Diagnostics

B.1 Evidence Distributions and Boundary Profiles

Data and aggregation. We analyze the same 1,600 π_0 RoboTwin2.0 episodes as Fig. 2: eight tasks, two settings, and 100 episodes per task–setting pair, with 291 successes and 1,309 failures. All episodes execute fixed $K = H = 50$. The candidates $k \in \{10, 20, 30, 40, 50\}$ are scored on these recorded traces. They are not five separate executed-horizon experiments.

For each candidate, we average finite scores over replans within an episode, then report medians and IQRs over episode means. The valid counts can differ between the two metrics because inter-chunk evidence requires executed history. The intra score follows Eq. 5, with prefixes zero-padded to H, a three-step baseline, and RMS over all ten denoising steps. Recomputed scores match the logs within 1.5×10^{-8}.

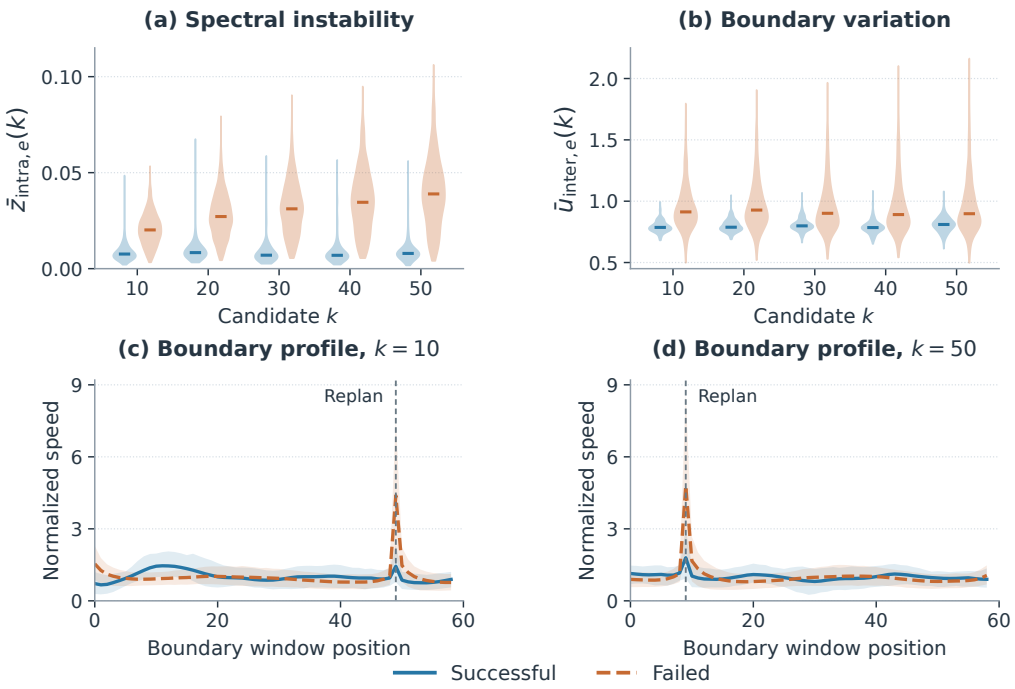

Figure 7: **Full evidence distributions and boundary behavior. (a,b)** Episode means of the operational scores, with violin marks indicating medians. Widths show within-group density, not relative sample counts. **(c,d)** Normalized speed profiles for candidate $k = 10$ and $k = 50$, with dashed vertical lines marking the history–prediction join. Curves show means and bands show P10–P90 across episode profiles, not confidence intervals or the IQR bands used in the main figure.

For boundary profiles, we compute first-difference speeds in the $W_h = 60$ stitched window of Eq. 6, normalize by the window mean, and average within each episode before summarizing across episodes. This prevents longer episodes receiving extra weight.

B.2 Joint Risk and Episode Outcomes

At the executed $K = 50$, each episode's intra/inter risk is the 75th percentile of $1 - q_{\text{intra},t}(50)$ or $1 - q_{\text{inter},t}(50)$ over valid replans. Pooled median thresholds are 0.98168436 and 0.96278395, respectively. Values strictly below the threshold are low and ties are high, so group sizes can differ.

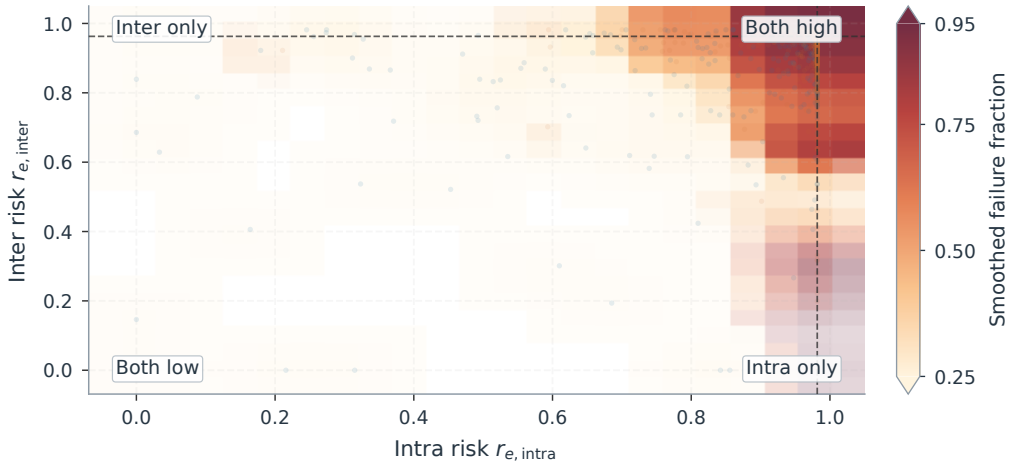

Figure 8: **Joint-risk landscape on the same 1,600 episodes.** Dashed lines mark the original median thresholds. Points denote episodes, with outcome colors as in Fig. 7. Color shows the smoothed local failure fraction. Lower opacity denotes lower smoothed occupancy. The color scale saturates below 0.25 and above 0.95. The map is descriptive. Group failure rates below are computed directly from episode outcomes, not read from the smoothed colors.

Table 6: **Exact median-split group outcomes.** Counts and failure rates reproduce Fig. 2(c).

Group	Intra risk	Inter risk	Episodes	Failures	Failure rate
Both low	Low	Low	313	157	50.2%
Inter only	Low	High	193	144	74.6%
Intra only	High	Low	487	417	85.6%
Both high	High	High	607	591	97.4%

The map uses a 23×23 grid with 7% range padding and separable kernel $[1, 4, 6, 4, 1]/16$. Success/failure counts are smoothed separately before division (stabilizer 10^{-12}). Opacity scales to the 90th occupancy percentile, hiding cells below 10^{-3}.

Scope of the evidence. These are pooled, retrospective associations from one policy and a fixed execution horizon. They do not establish causality, calibrated failure prediction, within-task effects independent of task difficulty, or how long a particular prefix remains reliable at the current replan. The benefit of adaptive execution is evaluated separately in the policy comparisons and ablations, not inferred from this heatmap.

C Full Simulation Benchmark Results

C.1 QHA Transfer to Held-out Tasks

For Tab. 2B, QHA is trained on Handover Block, Handover Mic, Hanging Mug, Place A2B Left, Place Bread Skillet, and Place Can Basket. Blocks Ranking RGB and Place Bread Basket are held out. The QHA head uses step 10,000, and frozen task-specific $\pi_{0.5}$ bases use step-20,000 clean50 checkpoints without quantile normalization. All selectors use $\{1, \ldots, 50\}$ and expected-round selection at temperature 1.0. AHS+QHA uses $\gamma = 1.0$. This cohort differs from the eight-task study in Tab. 2A.

AHS, QHA-only, and AHS+QHA share 100 episode identities per task and setting (400 per method). Historical Base in Tab. 2B is outside this paired cohort. The transfer comparison is AHS+QHA

versus AHS. Table 7 retains all four conditions: fusion improves two and reduces success in two. The aggregate gain does not establish uniform improvement or transfer of the base policy.

Table 7: **Complete held-out comparison.** Success rate (%) for the $\pi_{0.5}$ policy with QHA trained on six tasks and evaluated on two held-out tasks. Each task and setting uses 100 paired episodes. $\Delta = \mathrm{SR}_{\mathrm{AHS+QHA}} - \mathrm{SR}_{\mathrm{AHS}}$.

Task	Setting	AHS	QHA-only	AHS+QHA	Δ (pp)
Blocks Ranking RGB	Easy	44.00	48.00	**57.00**	+13.00
Blocks Ranking RGB	Hard	**33.00**	29.00	30.00	-3.00
Place Bread Basket	Easy	**52.00**	50.00	49.00	-3.00
Place Bread Basket	Hard	31.00	31.00	**35.00**	+4.00
Overall	Both	40.00	39.50	**42.75**	+2.75

Same-state decision disagreement. In AHS+QHA traces, the QHA and AHS component choices differ at 77.02% of 12,573 replans, with a mean absolute gap of 1.83 action steps. Easy contributes 5,533 replans (77.17%, 1.84 steps) and Hard 7,040 (76.90%, 1.82 steps). This measures differences at the same states, without establishing complementarity or explaining success gains.

C.2 Complete 50-task RoboTwin2.0 Evaluation

Table 8: **Complete 50-task RoboTwin2.0 results.** Success rate (%) for the shared multitask $\pi_{0.5}$ checkpoint, with 20 episodes per task, setting, and method. Δ is AHS minus Base, averaged over Easy and Hard.

Task	Easy		Hard		Δ (pp)
	Base	AHS	Base	AHS	
adjust bottle	100.00	100.00	90.00	95.00	+2.50
beat block hammer	70.00	80.00	15.00	50.00	+22.50
blocks ranking rgb	80.00	95.00	40.00	70.00	+22.50
blocks ranking size	25.00	45.00	25.00	35.00	+15.00
click alarmclock	60.00	65.00	45.00	45.00	+2.50
click bell	35.00	70.00	55.00	55.00	+17.50
dump bin bigbin	95.00	95.00	80.00	75.00	-2.50
grab roller	100.00	100.00	90.00	90.00	+0.00
handover block	65.00	65.00	10.00	35.00	+12.50
handover mic	100.00	100.00	15.00	15.00	+0.00
hanging mug	20.00	30.00	5.00	10.00	+7.50
lift pot	65.00	70.00	30.00	35.00	+5.00
move can pot	50.00	75.00	10.00	25.00	+20.00
move pillbottle pad	55.00	80.00	30.00	30.00	+12.50
move playingcard away	90.00	80.00	75.00	75.00	-5.00
move stapler pad	20.00	35.00	20.00	5.00	+0.00
open laptop	95.00	90.00	85.00	70.00	-10.00
open microwave	85.00	50.00	35.00	25.00	-22.50
pick diverse bottles	45.00	55.00	35.00	70.00	+22.50
pick dual bottles	55.00	70.00	75.00	75.00	+7.50
place a2b left	75.00	70.00	55.00	65.00	+2.50
place a2b right	70.00	75.00	50.00	50.00	+2.50
place bread basket	55.00	75.00	55.00	55.00	+10.00
place bread skillet	35.00	55.00	60.00	55.00	+7.50
place burger fries	95.00	95.00	90.00	100.00	+5.00
place can basket	40.00	85.00	0.00	35.00	+40.00
place cans plasticbox	40.00	85.00	55.00	80.00	+35.00
place container plate	75.00	90.00	80.00	80.00	+7.50
place dual shoes	75.00	85.00	50.00	60.00	+10.00
place empty cup	100.00	100.00	65.00	70.00	+2.50

Table 8 continued

Task	Easy		Hard		Δ (pp)
	Base	AHS	Base	AHS	
place fan	75.00	75.00	45.00	40.00	-2.50
place mouse pad	15.00	45.00	25.00	25.00	+15.00
place object basket	50.00	60.00	20.00	35.00	+12.50
place object scale	60.00	85.00	45.00	50.00	+15.00
place object stand	85.00	80.00	55.00	90.00	+15.00
place phone stand	65.00	65.00	45.00	35.00	-5.00
place shoe	85.00	90.00	75.00	55.00	-7.50
press stapler	100.00	90.00	60.00	70.00	+0.00
put bottles dustbin	45.00	65.00	50.00	55.00	+12.50
put object cabinet	25.00	35.00	20.00	15.00	+2.50
rotate qrcode	80.00	85.00	30.00	30.00	+2.50
scan object	20.00	40.00	20.00	50.00	+25.00
shake bottle	100.00	100.00	100.00	100.00	+0.00
shake bottle horizontally	100.00	100.00	100.00	100.00	+0.00
stack blocks three	75.00	55.00	35.00	45.00	-5.00
stack blocks two	85.00	100.00	80.00	90.00	+12.50
stack bowls three	60.00	65.00	35.00	30.00	+0.00
stack bowls two	100.00	85.00	85.00	90.00	-5.00
stamp seal	35.00	30.00	25.00	30.00	+0.00
turn switch	40.00	40.00	25.00	25.00	+0.00
Mean by setting	65.40	73.10	48.00	53.90	+6.80
Overall	Base: 56.70		AHS: 63.50		+6.80

Table 8 expands the multitask $\pi_{0.5}$ row of Tab. 1. Base and AHS succeed in 1,134 and 1,270 of 2,000 episodes, respectively, giving 56.70% and 63.50% overall. Easy success increases from 65.40% to 73.10%, and Hard from 48.00% to 53.90%. All 50 tasks are retained, including nine tasks with a negative change after averaging the two settings.

Statistical robustness and pairing scope. The full-matrix gain is 6.80 percentage points, with a task-cluster 95% bootstrap interval of $[3.75,\ 9.95]$. This interval resamples the 50 tasks, retaining both settings and methods within each task. Only 62 of 100 task–setting cells preserve exact seed-and-instruction pairing, while the remaining 38 cells use deterministic, outcome-blind, method-specific fallback identities after scene-construction failures. The 1,240 exact episode pairs yield a gain of 7.58 percentage points with a paired 95% interval of $[5.08,\ 10.08]$, resampling episodes within these cells. Both intervals use 10,000 bootstrap draws, but their statistical units and populations differ: the paired interval applies only to the exact-identity subset, not the full matrix.

C.3 Eight-task Policy-family Evaluation

The task-specific π_0 and $\pi_{0.5}$ matrices appear in Tab. 2A. Table 9 provides the corresponding Base–AHS breakdown for Fast-WAM and X-VLA (Zheng et al., 2026), evaluated on the same eight tasks with 100 rollouts per task and setting. Both use $\mathcal{K} = \{10, 20, 30\}$. Each Base–AHS comparison uses the same released policy checkpoint. Fast-WAM's overall success increases from 86.81% to 88.13%. For X-VLA, AHS slightly improves overall success from 47.44% to 47.75%: Easy increases from 74.50% to 75.75%, while Hard decreases from 20.38% to 19.75%. Individual task regressions are retained for both policies.

C.4 RoboCasa GR1 Tabletop

Table 10 reports the full 24-task RoboCasa GR1 Tabletop breakdown for $\pi_{0.5}$ and the three GR00T-family policies summarized in Tab. 1. QwenFAST (discrete tokens) and QwenPI (flow-matching action expert) are additional StarVLA-codebase references (Ye et al., 2026a; Community, 2026), both using Qwen3VL (Bai et al., 2025). Neither has an AHS counterpart in this table. For $\pi_{0.5}$, the 24-task means are 40.08% for Base and 42.50% for AHS, matching Tab. 1.

Table 9: **Fast-WAM and X-VLA per-task success rates on RoboTwin2.0.** Success rate (%) over 100 rollouts per task and setting. Easy and Hard denote clean and randomized settings. Each Base–AHS comparison fixes the policy checkpoint. Yellow columns use AHS, and bold marks the better method within each policy and setting, including ties.

Task	Fast-WAM (Yuan et al., 2026)				X-VLA (Zheng et al., 2026)			
	Base		AHS		Base		AHS	
	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Blocks Ranking RGB	99.00	98.00	**100.00**	**100.00**	87.00	34.00	**95.00**	**38.00**
Handover Block	**94.00**	81.00	91.00	**82.00**	**89.00**	**2.00**	88.00	1.00
Handover Mic	**100.00**	99.00	**100.00**	**100.00**	97.00	**1.00**	**100.00**	**1.00**
Hanging Mug	66.00	64.00	**71.00**	**69.00**	32.00	**8.00**	**35.00**	4.00
Place A2B Left	**95.00**	**95.00**	93.00	93.00	31.00	**25.00**	**40.00**	21.00
Place Bread Basket	91.00	91.00	**93.00**	**94.00**	**87.00**	41.00	80.00	**42.00**
Place Bread Skillet	91.00	91.00	**94.00**	**94.00**	86.00	22.00	**88.00**	**23.00**
Place Can Basket	**71.00**	63.00	69.00	**67.00**	**87.00**	**30.00**	80.00	28.00
Mean by setting	88.38	85.25	**88.88**	**87.38**	74.50	**20.38**	**75.75**	19.75
Overall	86.81		**88.13**		47.44		**47.75**	

Table 10: **Full per-task success rates on RoboCasa GR1 Tabletop.** Full 24-task breakdown, with 50 rollouts per task for $\pi_{0.5}$. Yellow columns use AHS and parentheses report percentage-point deltas over Base. Bold marks the best result in each row, including ties.

Task	QwenFAST +Qwen3VL	QwenPI +Qwen3VL	Isaac-GR00T N1.5		Isaac-GR00T N1.6		Qwen3GR00T +Qwen3VL		$\pi_{0.5}$	
			Base	AHS	Base	AHS	Base	AHS	Base	AHS
PnPBottleToCabinetClose	38.0	26.0	64.0	**68.0** (+4.0)	51.5	54.0 (+2.5)	46.0	64.0 (+18.0)	64.00	**68.00** (+4.00)
PnPCanToDrawerClose	44.0	62.0	18.0	12.0 (-6.0)	13.0	12.0 (-1.0)	**80.0**	**80.0** (+0.0)	58.00	56.00 (-2.00)
PnPCupToDrawerClose	**56.0**	42.0	12.0	4.0 (-8.0)	8.5	14.0 (+5.5)	54.0	52.0 (-2.0)	34.00	40.00 (+6.00)
PnPMilkToMicrowaveClose	44.0	**50.0**	38.0	34.0 (-4.0)	14.0	20.0 (+6.0)	48.0	42.0 (-6.0)	40.00	44.00 (+4.00)
PnPPotatoToMicrowaveClose	14.0	42.0	**54.0**	36.0 (-18.0)	41.5	50.0 (+8.5)	28.0	28.0 (+0.0)	22.00	30.00 (+8.00)
PnPWineToCabinetClose	14.0	32.0	16.0	20.0 (+4.0)	16.5	24.0 (+7.5)	46.0	52.0 (+6.0)	52.00	**56.00** (+4.00)
PnPNovelFromCuttingboardToBasket	54.0	40.0	50.0	52.0 (+2.0)	58.0	54.0 (-4.0)	48.0	**70.0** (+22.0)	34.00	34.00 (+0.00)
PnPNovelFromCuttingboardToCardboardbox	42.0	46.0	36.0	34.0 (-2.0)	46.5	46.0 (-0.5)	40.0	**54.0** (+14.0)	32.00	40.00 (+8.00)
PnPNovelFromCuttingboardToPan	58.0	60.0	68.0	64.0 (-4.0)	68.5	**80.0** (+11.5)	68.0	**80.0** (+12.0)	54.00	58.00 (+4.00)
PnPNovelFromCuttingboardToPot	58.0	40.0	34.0	56.0 (+22.0)	65.0	64.0 (-1.0)	52.0	**76.0** (+24.0)	46.00	40.00 (-6.00)
PnPNovelFromCuttingboardToTieredbasket	40.0	44.0	46.0	32.0 (-14.0)	46.5	54.0 (+7.5)	**56.0**	44.0 (-12.0)	22.00	28.00 (+6.00)
PnPNovelFromPlacematToBasket	36.0	44.0	50.0	46.0 (-4.0)	**58.5**	48.0 (-10.5)	42.0	54.0 (+12.0)	42.00	46.00 (+4.00)
PnPNovelFromPlacematToBowl	38.0	52.0	50.0	62.0 (+12.0)	57.5	60.0 (+2.5)	44.0	**66.0** (+22.0)	34.00	32.00 (-2.00)
PnPNovelFromPlacematToPlate	42.0	50.0	62.0	66.0 (+4.0)	63.0	**82.0** (+19.0)	48.0	72.0 (+24.0)	46.00	48.00 (+2.00)
PnPNovelFromPlacematToTieredshelf	18.0	28.0	14.0	26.0 (+12.0)	28.5	**36.0** (+7.5)	18.0	20.0 (+2.0)	28.00	28.00 (+0.00)
PnPNovelFromPlateToBowl	52.0	52.0	58.0	58.0 (+0.0)	57.0	58.0 (+1.0)	**60.0**	**60.0** (+0.0)	40.00	44.00 (+4.00)
PnPNovelFromPlateToCardboardbox	30.0	40.0	40.0	48.0 (+8.0)	43.5	**58.0** (+14.5)	50.0	54.0 (+4.0)	28.00	30.00 (+2.00)
PnPNovelFromPlateToPan	48.0	36.0	44.0	48.0 (+4.0)	51.0	**68.0** (+17.0)	54.0	54.0 (+0.0)	32.00	36.00 (+4.00)
PnPNovelFromPlateToPlate	50.0	48.0	66.0	74.0 (+8.0)	78.7	**82.0** (+3.3)	70.0	74.0 (+4.0)	54.00	54.00 (+0.00)
PnPNovelFromTrayToCardboardbox	28.0	34.0	44.0	52.0 (+8.0)	51.5	48.0 (-3.5)	38.0	**56.0** (+18.0)	48.00	50.00 (+2.00)
PnPNovelFromTrayToPlate	34.0	64.0	50.0	60.0 (+10.0)	**71.0**	68.0 (-3.0)	56.0	62.0 (+6.0)	40.00	40.00 (+0.00)
PnPNovelFromTrayToPot	46.0	44.0	46.0	50.0 (+4.0)	64.5	64.0 (-0.5)	50.0	**66.0** (+16.0)	54.00	54.00 (+0.00)
PnPNovelFromTrayToTieredbasket	36.0	50.0	44.0	38.0 (-6.0)	**57.0**	56.0 (-1.0)	36.0	56.0 (+20.0)	34.00	36.00 (+2.00)
PnPNovelFromTrayToTieredshelf	16.0	28.0	34.0	38.0 (+4.0)	31.5	34.0 (+2.5)	16.0	**44.0** (+28.0)	24.00	28.00 (+4.00)
Average	39.00	43.92	43.25	44.92 (+1.67)	47.61	51.42 (+3.80)	47.83	**57.50** (+9.67)	40.08	42.50 (+2.42)

C.5 Horizon-selection Comparisons

Table 11 reports additional controls for horizon selection, separately from the cross-policy results in Tab. 1.

Fixed, action-only, and evidence-update selectors. The π_0 study uses Place A2B Left, Place Bread Basket, Place Bread Skillet, and Place Can Basket under Easy and Hard settings, with 16 paired episodes per task and setting (128 per selector); the Instantaneous row is a separate historical reference. This is a separate evaluation cohort from the 100-rollout studies in Tabs. 3 and 4. The global fixed horizon $K = 20$ is selected retrospectively from historical results, not from a held-out validation set. Jerk-min uses an action-only smoothness criterion on $\mathcal{K} = \{10, 20, 30, 40, 50\}$.

The evidence-update variants share this candidate grid and the same instantaneous evidence, using expected-round selection with temperature 0.8. Instantaneous uses $q_{\mathrm{mix},t}(k)$, whereas EMA maintains $m_t(k) = \rho_{\mathrm{EMA}} m_{t-1}(k) + (1 - \rho_{\mathrm{EMA}}) q_{\mathrm{mix},t}(k)$ with $\rho_{\mathrm{EMA}} = 0.99$. Neither comparator uses Beta counts or kernel neighborhood sharing. Full Beta is the shared AHS reference. Results describe success and call-count trade-offs without exact compute matching. The paired 95% SR-difference interval between AHS and Jerk-min includes zero, so this compact study does not establish an SR advantage.

Table 11: **Horizon-selection outcomes and recorded costs.** RoboTwin2.0 uses π_0 on four tasks under Easy and Hard (128 episodes per selector). Instantaneous is historical, with matching timing unavailable. Inference and wall times are seconds per episode. RoboCasa references are non-paired. Bold/underline mark best/second-best SR among the displayed methods within each benchmark.

RoboTwin2.0

Selector	SR (%) ↑	Calls/ep	Policy infer	Total infer	Wall
Global fixed ($K = 20$)	<u>14.84</u>	25.96	2.75	2.75	54.33
Jerk-min	14.06	21.14	2.28	2.28	52.01
EMA q_{mix}	10.94	19.72	2.05	2.07	46.72
AHS (Full Beta)	**15.63**	20.71	2.26	2.28	53.47
Instantaneous q_{mix} (historical)	13.28	19.58	—	—	—

RoboCasa GR1 Tabletop, Qwen3GR00T, 24 tasks

	Base	AAC-core	AHS
SR (%) ↑	47.83	<u>52.58</u>	**57.50**

AAC-core on RoboCasa. We evaluate a joint-space adapter of Adaptive Action Chunking (Liang et al., 2026) on Qwen3GR00T over 24 tasks with 50 rollouts per task. Each replan draws 20 action-head samples under the same observation and instruction. A prefix-entropy elbow and a movement guard determine $K \in \{2, \ldots, 16\}$, and the first sampled chunk supplies the executed actions. The adapter operates on the policy's native 29-dimensional absolute joint targets. It is not an exact reproduction of the published Cartesian-action implementation. AAC-core obtains 52.58% success. Historical Base/AHS values from Tab. 1 provide non-paired context only. We do not report a paired difference or infer compute equivalence from policy call counts.

D Runtime and Latency

D.1 Policy-call and Runtime Costs

We group runtime measurements by available timing scope. Means include all episodes, including failures. Policy inference excludes separately timed selector work, but recorded total inference includes it. Episode wall time includes simulation and within-episode overhead, not physical robot execution time. These are descriptive profiles, not hardware-matched cross-policy rankings.

Compact-selector timings are included with their success rates in Tab. 11.

Shared timing scope for $\pi_{0.5}$. Table 12 reports policy inference rather than total inference: the 50-task logs lack a separate total, and QHA selection occurs outside the policy timer without a separate selector timer. The local AHS reference has a recorded total of 3.49 s per episode. Both QHA variants make fewer calls but have higher policy-inference and wall times than this reference. We keep the local profile separate from Tab. 2B and do not derive success-normalized efficiency across result versions. These costs characterize the recorded implementation, not an intrinsic QHA cost.

AAC-core sampling cost. On RoboCasa GR1 Tabletop with Qwen3GR00T (24 tasks, 1,200 episodes), AAC-core averages 180.95 calls, 20.98 s of synchronized total inference, and 47.56 s of

Table 12: **Policy-inference and episode costs for $\pi_{0.5}$.** RoboTwin2.0 averages include failures. The two panels are separate cohorts. Panel B profiles local QHA runs, not the success-rate version in Tab. 2B.

Method	Calls/ episode	Policy inference (s/episode)	Episode wall time (s/episode)
A. Full 50-task evaluation	2,000 episodes per method		
Base	8.00	0.97	38.19
AHS	14.50	1.55	36.83
B. Local QHA profile	2 held-out tasks, 400 episodes per method		
AHS	33.77	3.29	68.60
QHA-only	29.81	18.87	81.53
AHS+QHA	31.43	19.88	83.32

wall time per episode. Each call samples 20 action chunks, giving 4,342,740 chunks in total. No policy-only timing aggregate is available. Historical Base/AHS references are not joined to these records. RTX 4090 use is documented for AAC-core and the 50-task study. Exact device metadata is unverified for the compact and QHA bundles.

D.2 Latency and Asynchronous Execution

The policy prediction horizon is $H = 50$. Fixed $K = 40$ is a target execution budget. RTC can replace a chunk at the first legal waypoint boundary after readiness. The AHS candidates are $\{10, 20, 30, 40\}$. This cohort uses no QHA and is separate from the $K = 50$ candidate-scaling experiment.

Timing and aggregation. For episode i, let $a_{ij} = t_{ij}^{\text{start}} - t_{ij}^{\text{obs}}$ be the age of the observation used to generate executed action j. We compute $\bar{a}_i = n_i^{-1} \sum_j a_{ij}$ and average $\bar{a}_i$ equally over episodes within each task, then equally over the four tasks, separately for Easy and Hard. Failures are included.

Wait s/ep averages recorded hold time per episode. Wait % is $100 \sum_i W_i / \sum_i T_i$, where T_i includes both action execution and waiting. Calls/ep includes requests discarded at episode termination. Infer s/ep reports model computation time per episode, amortized across active batch requests and excluding selector computation.

Table 13: **Latency results by task and in aggregate on RoboTwin2.0 Hard.** 100 paired episodes per task/method/delay, including failures. Costs and mean observation age use +200 ms. Fixed uses target $K = 40$. Bold/underline mark best/second-best SR, waiting, and age within each group, including displayed ties. Calls and inference time are unranked.

Method	SR (%) ↑			Wait (%) ↓	Wait s/ep ↓	Calls/ep	Infer s/ep	Mean age (s) ↓
	+0 ms	+100 ms	+200 ms					
All four tasks								
Sync-Fixed	19.50	19.50	19.25	3.77	4.432	12.89	5.04	5.02
Sync-AHS	26.25	24.25	23.75	6.80	7.893	22.95	8.59	**3.00**
RTC-Fixed	23.00	21.50	22.00	**0.31**	**0.344**	13.32	6.47	4.92
RTC-AHS	**26.50**	**25.75**	**26.75**	**0.31**	**0.344**	24.30	10.62	3.07
Place A2B Left								
Sync-Fixed	15.00	21.00	16.00	3.37	3.106	9.03	3.29	5.35
Sync-AHS	20.00	16.00	20.00	5.92	5.790	16.83	6.21	**3.30**
RTC-Fixed	**26.00**	**22.00**	**22.00**	0.41	**0.344**	9.53	4.09	5.15
RTC-AHS	22.00	21.00	**22.00**	**0.37**	**0.344**	17.94	7.28	3.37
Place Bread Basket								
Sync-Fixed	26.00	25.00	25.00	3.20	5.246	15.25	6.12	5.77
Sync-AHS	**38.00**	**34.00**	32.00	6.19	9.409	27.36	10.14	**3.20**
RTC-Fixed	28.00	29.00	29.00	**0.22**	**0.344**	15.61	7.80	5.68
RTC-AHS	37.00	31.00	**37.00**	0.24	**0.344**	28.68	12.60	3.38
Place Bread Skillet								
Sync-Fixed	13.00	9.00	12.00	5.15	4.097	11.91	4.51	3.89
Sync-AHS	13.00	13.00	12.00	8.56	6.994	20.33	7.06	2.58
RTC-Fixed	13.00	12.00	10.00	**0.45**	0.346	12.35	5.59	3.81
RTC-AHS	**15.00**	**16.00**	**13.00**	**0.45**	**0.345**	21.71	8.43	**2.54**
Place Can Basket								
Sync-Fixed	24.00	23.00	24.00	3.90	5.280	15.35	6.27	5.08
Sync-AHS	**34.00**	34.00	31.00	7.07	9.381	27.27	10.95	**2.93**
RTC-Fixed	25.00	23.00	27.00	**0.27**	**0.344**	15.78	8.39	5.02
RTC-AHS	32.00	**35.00**	**35.00**	**0.27**	**0.344**	28.86	14.19	3.01

Success within a time budget. For T_i^{success}, set the recorded completion time for a successful episode and $+\infty$ for a failed episode. Then

$$\mathrm{SR}(t) = \frac{100}{4} \sum_{q=1}^{4} \frac{1}{100} \sum_{i \in q} \mathbf{1}\{T_i^{\text{success}} \leq t\}.$$

Here t denotes physical time in seconds. Computed separately for each setting, $\mathrm{SR}(t)$ measures task completion under a time budget, with all attempted episodes retained in the denominator.

Easy and Hard outcomes. Both settings use the same task checkpoints, execution horizon and AHS candidates; episodes are paired across methods and delays within each setting. At +200 ms, RTC reduces AHS waiting by 94.5% on Easy and 95.6% on Hard. Relative to RTC-Fixed, RTC-AHS achieves higher aggregate SR in all six setting/delay conditions, with observed gains of 3.50–8.25 percentage points. At +200 ms, the paired 95% interval for this gain is $[-0.75, 8.26]$ points on Easy and $[0.75, 8.75]$ on Hard. Relative to Sync-AHS, RTC-AHS changes final SR by -2.00 points on Easy and $+3.00$ points on Hard at +200 ms. Thus, reduced waiting does not imply uniformly higher success. Policy calls and model computation remain higher than for RTC-Fixed (Tabs. 13, 14, and 15).

Table 14: **Latency results by task and in aggregate on RoboTwin2.0 Easy.** 100 paired episodes per task/method/delay, including failures. Costs and mean observation age use +200 ms. Fixed uses target $K = 40$. Bold/underline mark best/second-best SR, waiting, and age within each group, including displayed ties. Calls and inference time are unranked.

Method	SR (%) ↑			Wait (%) ↓	Wait s/ep ↓	Calls/ep	Infer s/ep	Mean age (s) ↓
	+0 ms	+100 ms	+200 ms					
All four tasks								
Sync-Fixed	38.00	35.50	37.00	3.90	3.856	11.21	4.38	5.12
Sync-AHS	**46.00**	46.25	**44.75**	6.63	6.310	18.35	6.90	**3.16**
RTC-Fixed	38.75	40.25	39.00	**0.37**	**0.344**	11.50	5.57	5.00
RTC-AHS	**46.00**	**48.50**	42.75	**0.37**	**0.344**	20.32	8.80	3.18
Place A2B Left								
Sync-Fixed	49.00	45.00	50.00	3.37	2.401	6.98	2.48	5.65
Sync-AHS	**54.00**	**58.00**	56.00	5.86	4.107	11.94	4.25	**3.41**
RTC-Fixed	50.00	51.00	50.00	**0.51**	**0.344**	7.55	3.36	5.44
RTC-AHS	52.00	57.00	**57.00**	**0.51**	**0.344**	12.83	5.21	3.51
Place Bread Basket								
Sync-Fixed	40.00	35.00	35.00	3.41	4.548	13.22	5.29	5.51
Sync-AHS	**47.00**	47.00	**49.00**	6.10	7.317	21.27	8.19	**3.28**
RTC-Fixed	41.00	42.00	41.00	**0.28**	**0.344**	13.19	6.56	5.48
RTC-AHS	46.00	**48.00**	41.00	0.29	**0.344**	24.49	10.41	3.33
Place Bread Skillet								
Sync-Fixed	26.00	26.00	27.00	4.98	3.643	10.59	4.02	4.25
Sync-AHS	36.00	33.00	29.00	7.86	5.853	17.02	6.00	**2.81**
RTC-Fixed	25.00	31.00	21.00	**0.48**	**0.345**	11.41	5.18	4.02
RTC-AHS	**39.00**	**39.00**	**34.00**	0.51	**0.345**	17.39	7.00	2.90
Place Can Basket								
Sync-Fixed	37.00	36.00	36.00	4.12	4.832	14.05	5.75	5.09
Sync-AHS	**47.00**	47.00	**45.00**	6.84	7.964	23.15	9.16	3.12
RTC-Fixed	39.00	37.00	44.00	0.33	**0.344**	13.83	7.17	5.06
RTC-AHS	**47.00**	**50.00**	39.00	**0.30**	**0.344**	26.57	12.60	**3.00**

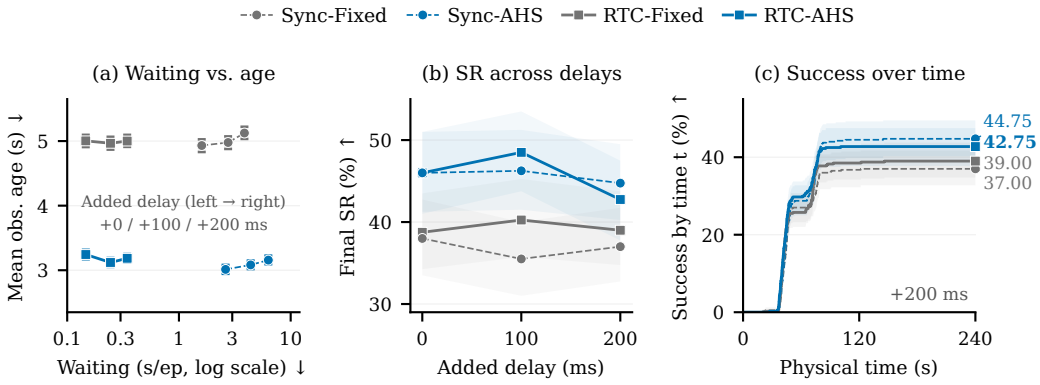

Figure 9: **AHS with asynchronous execution on $\pi_{0.5}$ Easy.** The three panels mirror Fig. 5: waiting versus mean observation age at +0/+100/+200 ms, final SR at each delay, and SR(t) at +200 ms. All 400 outcomes per condition are included. Bars and shading show marginal/pointwise 95% paired-seed bootstrap intervals within four tasks.

Table 15: **Aggregate costs at +0 and +100 ms on Easy and Hard.** Each method/delay has 400 outcomes per setting, including failures. The +200 ms aggregates appear with the per-task results in Tabs. 13 and 14. Observation age averages within episodes, then equally across episodes and tasks. Within each setting and delay, best and second-best distinct waiting and age values are bold and underlined. Displayed ties share a mark. Calls/ep and Infer s/ep are descriptive quantities and are not ranked.

Delay (ms)	Method	Wait (%) ↓	Wait s/ep ↓	Calls/ep	Infer s/ep	Mean age (s) ↓
Easy						
0	Sync-Fixed	1.67	1.600	11.11	4.32	4.93
	Sync-AHS	2.87	2.614	18.15	6.78	**3.01**
	RTC-Fixed	**0.16**	**0.147**	11.20	5.38	5.00
	RTC-AHS	**0.16**	**0.147**	17.86	7.81	3.24
100	Sync-Fixed	2.82	2.755	11.29	4.37	4.98
	Sync-AHS	4.80	4.384	17.97	6.60	**3.08**
	RTC-Fixed	**0.26**	**0.244**	11.42	5.45	4.97
	RTC-AHS	0.28	**0.244**	18.73	8.09	3.12
Hard						
0	Sync-Fixed	1.60	1.853	12.87	5.14	4.84
	Sync-AHS	2.93	3.244	22.53	8.37	**2.85**
	RTC-Fixed	**0.13**	**0.146**	12.84	6.31	4.99
	RTC-AHS	0.14	0.147	22.23	9.83	3.01
100	Sync-Fixed	2.72	3.168	12.99	5.12	4.89
	Sync-AHS	4.85	5.523	22.64	8.39	**2.93**
	RTC-Fixed	**0.22**	**0.244**	13.34	6.64	4.84
	RTC-AHS	**0.22**	**0.244**	24.05	10.49	2.98

E AHS Hyperparameter Sensitivity

We evaluate $\pi_{0.5}$ + AHS on eight RoboTwin2.0 tasks under Easy and Hard settings, with $\mathcal{K} = \{10, 20, 30, 40, 50\}$ and expected-round selection at $T_{\text{sel}} = 0.8$. The two studies below use different episode budgets and pairing protocols. Neither changes our default configuration.

Inter-chunk continuity window. We vary only $W_h \in \{30, 60, 90\}$, sharing 16 episode identities per task and setting (256 per configuration). The default $W_h = 60$ is evaluated afresh as the paired reference in Fig. 10.

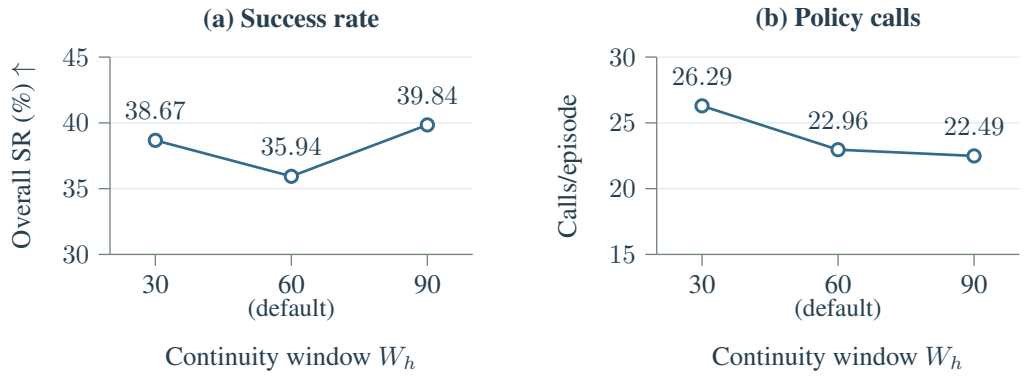

Figure 10: **Inter-chunk continuity-window sensitivity.** $\pi_{0.5}$ + AHS on eight RoboTwin2.0 tasks, Easy and Hard, with 256 matched episodes per configuration. Lines connect tested points. Paired difference intervals appear in the text.

Compared with $W_h = 60$, gains for 30 and 90 are $+2.73$ and $+3.91$ percentage points, with paired 95% confidence intervals $[-2.34, 7.81]$ and $[-0.78, 8.59]$. We use 10,000 bootstrap draws of matched episodes within task–setting cells, weighted equally. Both intervals include zero, so neither a success-rate difference nor equivalence is established. We retain $W_h = 60$.

Other evidence and posterior hyperparameters. Each configuration in Tab. 16 uses 100 episodes per task and setting (1,600 total), unpaired across configurations. One parameter family changes at a time, with the two penalties varied jointly. Other settings follow Appendix A. These historical runs are separate from the window study. Overall SR spans 34.38–37.13%, characterizing sensitivity rather than validation-based selection or an established optimum.

Table 16: **Sensitivity to other AHS hyperparameters.** Success rates (%) on eight RoboTwin2.0 tasks, with 1,600 episodes per configuration. Bold/underline mark higher/lower SR within each parameter family. Defaults specify parameters, not additional evaluations.

Parameter	Default	Tested	Easy ↑	Hard ↑	Overall ↑
Forgetting ρ_f	0.99	0.95	**46.88**	**26.75**	**36.81**
		0.995	<u>46.50</u>	<u>25.88</u>	<u>36.19</u>
Update strength η	1	0.5	<u>45.38</u>	<u>24.38</u>	<u>34.88</u>
		2	**47.38**	**26.88**	**37.13**
Kernel bandwidth σ_K	10	5	**45.50**	**26.13**	**35.81**
		20	<u>44.25</u>	<u>24.50</u>	<u>34.38</u>
Intra reference window W_τ	3	1	**47.38**	**24.75**	**36.06**
		5	<u>46.38</u>	<u>23.88</u>	<u>35.13</u>
Joint $\alpha_{\text{intra}} = \beta_{\text{inter}}$	4	2	**46.63**	<u>23.63</u>	<u>35.13</u>
		8	<u>46.25</u>	**24.88**	**35.56**

F More Case Studies

Handover Block. Fig. 11 illustrates phase-dependent execution in a successful π_0 rollout. AHS selects shorter horizons during grasping and transport phases that require closed-loop correction, and longer horizons once the motion stabilizes. This example illustrates the behavior of the online selector in Sec. 4.1. It is not a controlled comparison of posterior mechanisms.

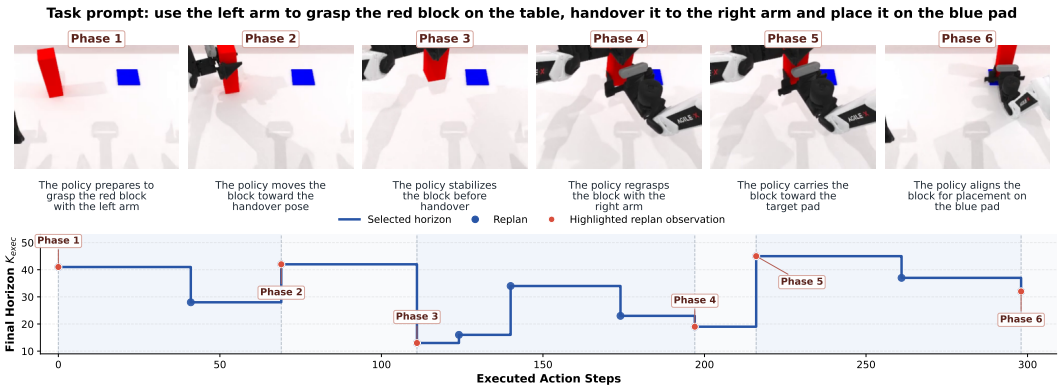

Figure 11: π_0 **Handover Block case study.** The selected execution horizon changes with task phase in a successful RoboTwin2.0 rollout. K_{exec} denotes the selected K_t.

Additional tasks and settings. Fig. 12 shows four additional π_0 RoboTwin2.0 rollouts across Easy and Hard settings. Fig. 13 adds $\pi_{0.5}$+AHS cases on four real-world tasks. These qualitative examples illustrate horizon changes across task phases, not additional scored trials.

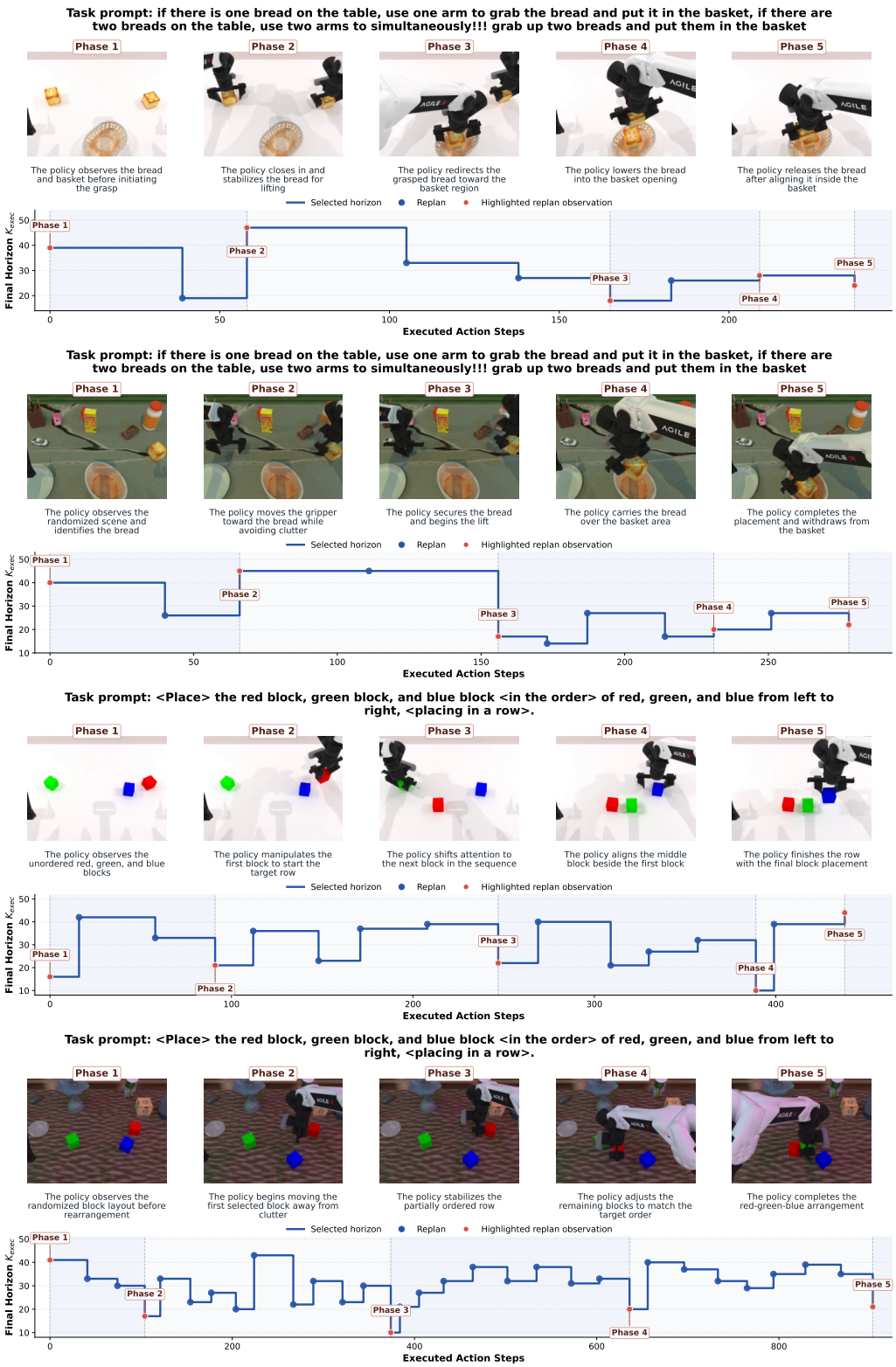

Figure 12: **Additional π_0 case studies on RoboTwin2.0.** Four successful rollouts show phase-dependent AHS horizons across Place Bread Basket and Blocks Ranking RGB under Easy and Hard settings.

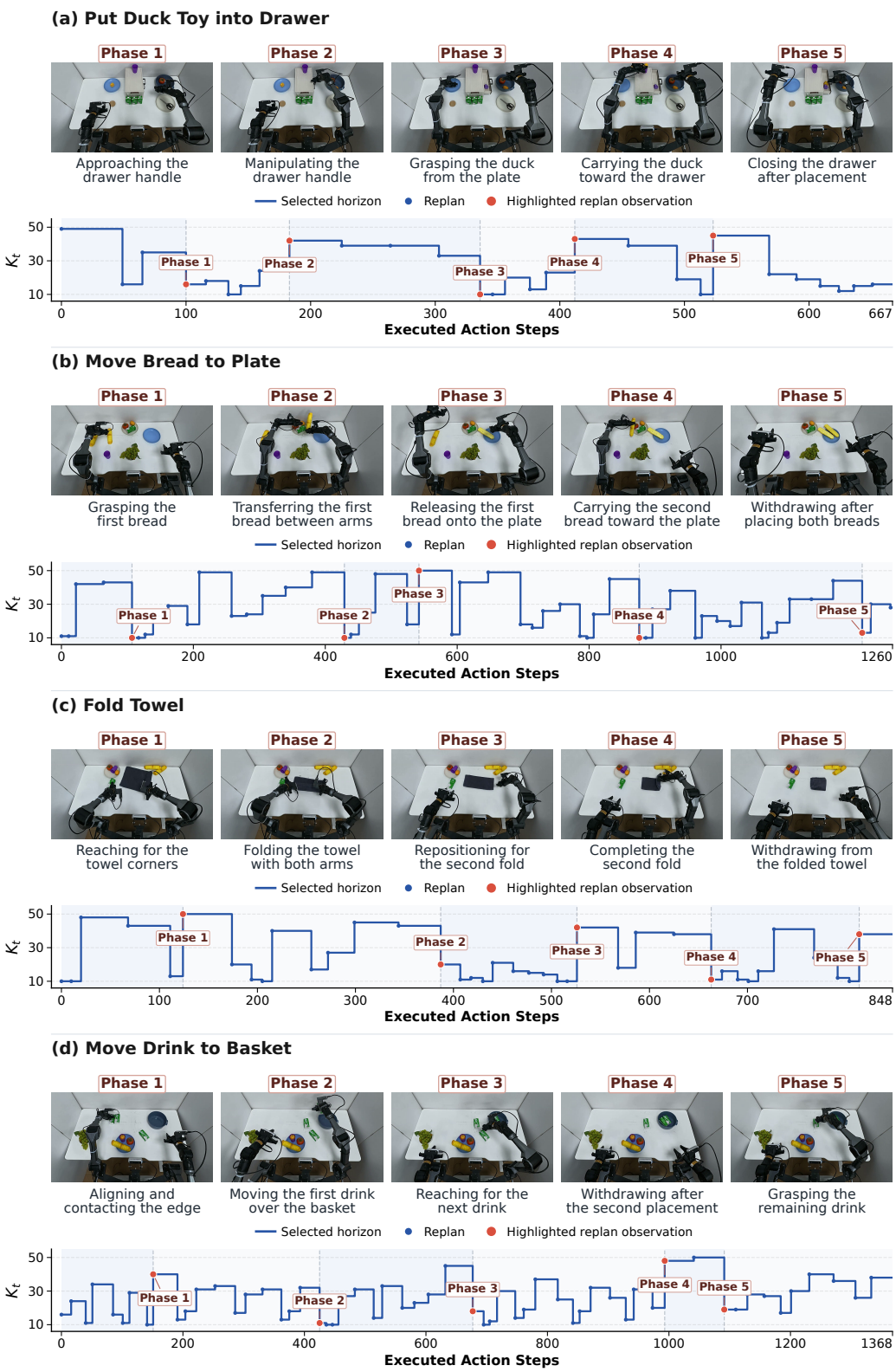

Figure 13: **Real-world $\pi_{0.5}$+AHS case studies.** Highlighted observations are selected at large adjacent-replan changes in K_t. Curves retain every logged horizon through the recorded endpoint, without smoothing. Phase labels are manual visual annotations, not ground-truth boundaries or trial scores.

G Discussion and Limitations

Additional background. RT-1, RT-2, and PaLM-E connect large-scale learning with robot control (Brohan et al., 2023; Zitkovich et al., 2023; Driess et al., 2023). Open X-Embodiment, Octo, and OpenVLA extend shared data and generalist initialization (O'Neill et al., 2024; Ghosh et al., 2024; Kim et al., 2024), while RoboBrain 2.0 broadens embodied reasoning interfaces (Team et al., 2025). RDT-1B and DexVLA extend diffusion-based action generation (Liu et al., 2025a; Wen et al., 2025). Other approaches couple actions to world models (Bi et al., 2025) or adapt chunking through self-guidance (So et al., 2026). ChunkTrust adapts chunk execution using action-expert evidence.

Limitations. AHS requires access to action chunks and generation-time velocity traces. It scores a candidate grid, although expected-round selection can return intermediate integer lengths. QHA learns a dense prior, but AHS+QHA still computes online evidence, interpolated or scored directly on the dense grid. Our evaluations cover multiple policies and two simulation benchmarks, plus four real-world bimanual tasks, rather than all embodiments, safety-critical tasks, or long-horizon mobile manipulation.

Broader impact. Adaptive horizons may reduce unnecessary replanning while preserving reactivity near contact. An incorrect horizon can still commit a robot to unsafe motion outside the tested distribution. Deployment should retain workspace and speed limits, emergency stops, human supervision during evaluation, and task-specific validation.

Existing assets. We use the cited RoboTwin2.0 and RoboCasa benchmarks, policy checkpoints, and associated codebases for research evaluation. These assets are not repackaged in this draft. Released derivatives will retain the licenses, terms, and attribution required by the upstream projects.