TEMPORAL ENSEMBLING REVISITED: RECOVERING THE STOCHASTIC HEADROOM OF GENERATIVE ROBOT POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion and flow-matching policies formulate robot control as a generative process, decoding each action chunk from sampled noise. Under closed-loop deployment, a failed rollout may therefore reflect an unfavorable sequence of noise draws rather than an inherent limitation of the policy, yet standard success rates leave the two indistinguishable. We examine this ambiguity with an episode-level pass@N probe that holds the task initialization fixed and varies only the policy noise. On π_0 LIBERO-Long, success rises from 80.7% with a single draw to 96.1% within eight, indicating that a substantial portion of long-horizon failure is sampling-induced. We recover this stochastic headroom online with Recency-weighted Temporal Ensembling (RTE), which exploits the overlapping predictions inherent to receding-horizon control, averaging them under weights that decay with prediction age. We show analytically that this weighting trades stale-plan bias against prediction noise: when predictions scatter around a plan that drifts with age, neither uniform averaging nor freshest-only execution minimizes per-step action error, and the error-minimizing weights decay with age. With a single fixed decay rate, RTE improves four flow-matching and diffusion backbones in most settings across LIBERO, RoboTwin 2.0, SIMPLER, and a real Franka arm. Controls that replan at matched strides confirm that temporal aggregation accounts for most of these gains, which replanning frequency alone cannot match, and that its contribution tracks the effective ensemble size.

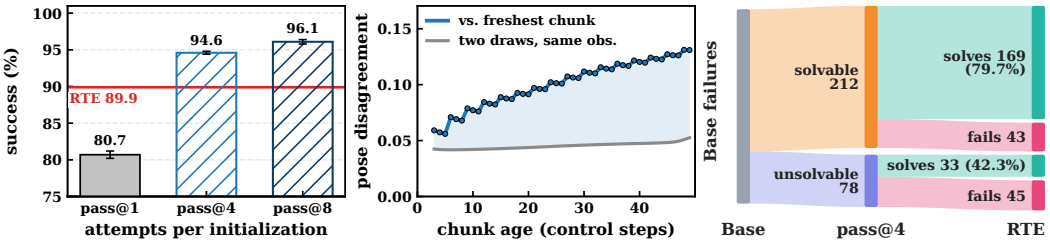
1 INTRODUCTION

Vision-language-action (VLA) models have demonstrated remarkable success in mapping natural language instructions and raw visual observations directly to low-level control across diverse manipulation tasks (Brohan et al., 2022; Zitkovich et al., 2023; Ghosh et al., 2024; Kim et al., 2025b;a; Li et al., 2026c). Much of this capability comes from generative action heads, which decode entire action chunks from sampled noise with diffusion or flow matching (Ho et al., 2020; Lipman et al., 2023; Chi et al., 2023; Li et al., 2024; Liu et al., 2025a; Black et al., 2024; 2025a; Bjorck et al., 2025) and thereby achieve the precision and multimodal coverage that dexterous manipulation requires.

However, this reliance on generative sampling renders deployment inherently stochastic. Given the same observation, the policy synthesizes different action chunks under different noise draws, and receding-horizon control (Mayne & Michalska, 1990) compounds this stochasticity over repeated queries into the executed trajectory. Consequently, a failed closed-loop rollout introduces a critical ambiguity: does the policy genuinely lack the necessary behavior for the task, or did it simply suffer from an unlucky sequence of noise samples? A conventional success rate metric fails to distinguish between these scenarios (Vincent et al., 2024; Kress-Gazit et al., 2024). Yet, isolating the cause is vital, as mitigating intrinsic policy shortcomings requires costly interventions such as fine-tuning (Kim et al., 2025a) or reinforcement learning (Li et al., 2026b; Chen et al., 2025a).

To formalize this distinction, we adapt the pass@N metric prevalent in test-time compute for language models (Kulal et al., 2019; Chen et al., 2021; Brown et al., 2024; Yue et al., 2025). By replaying a fixed task initialization under an identical environment and policy, varying only the noise seed, we measure how often the frozen policy succeeds when nothing but its noise changes. As shown in

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.



(a) pass@N on LIBERO-Long. (b) Plan drift vs. sampling noise. (c) Failures that RTE recovers.

Figure 1: **The diagnostic chain on LIBERO.** (a) Repeated noise draws expose a large stochastic headroom, raising success from 80.7% to 96.1% within eight draws; the red line marks RTE at $s{=}3$. (b) Over $1{,}500$ π_0 rollouts of RTE at $s{=}3$ on LIBERO-Long, an older chunk disagrees more with the freshest one as it ages, whereas two draws on the same observation stay at a nearly constant level; disagreement is the RMS pose gap, and shading marks the excess over sampling. (c) Of the 290 Base failures on π_0-Long, RTE at $s{=}3$ solves 79.7% of those that one of four resampled attempts solves, but only 42.3% of the rest.

Figure 1a, success on π_0 LIBERO-Long (Liu et al., 2023) escalates from 80.7% with a single draw to 96.1% within eight draws. This stark contrast, which we define as the *stochastic headroom*, indicates that a significant fraction of failures stems from sampling variance rather than capability deficits. While informative, this offline diagnostic assumes an oracle that can identify success in hindsight, a privilege unavailable during online robot deployment.

Fortunately, receding-horizon control naturally provides a mechanism for test-time sample aggregation (Wang et al., 2023). Because the policy replans before its previous chunk runs out (Chi et al., 2023), every control step receives several overlapping predictions, each from a different observation and noise draw. These predictions are not identically distributed: a fresher chunk is conditioned on the most recent observation, whereas a staler chunk relies on a diverging open-loop projection (Liu et al., 2025b). Averaging over more chunks therefore reduces prediction noise but admits stale-plan bias. We combine them with Recency-weighted Temporal Ensembling (RTE), which aligns the active chunks by target step on the client side and averages them under weights that decay exponentially with prediction age. Exponential ensembling of this kind dates back to ACT (Zhao et al., 2023), where the decay rate serves as a smoothing knob; we show that it sets a trade-off between stale-plan bias and prediction noise, and we find that it should decay with physical age rather than per query, so that one rate carries over across replanning strides. As shown in Figure 1b, plan disagreement grows with prediction age well beyond the level that sampling alone produces, and we prove that under such drift the error-minimizing weights decay with age.

We evaluate RTE with four diffusion and flow-matching backbones on LIBERO (Liu et al., 2023), SIMPLER (Li et al., 2025), RoboTwin 2.0 (Chen et al., 2025b), and a Franka arm, and compare it with freshest-only and uniform controls that replan at the same stride. Replanning more often without aggregation frequently lowers success, whereas blending the overlapping chunks by age turns the extra queries into gains, mostly by recovering failures that resampling can fix, and matters most when the blended predictions are old, as with long chunks.

In summary, our core contributions are:

- We formulate *stochastic headroom* via an episode-level pass@N diagnostic, revealing that a substantial portion of policy failure stems from sampling stochasticity rather than capability limits.
- We propose *Recency-weighted Temporal Ensembling* (RTE), a training-free, client-side estimator that recovers much of this headroom online. We prove that under plan drift, uniform averaging and freshest-only execution are both suboptimal, while the error-minimizing weights decay with prediction age.
- Across four backbones, three benchmarks, and a Franka robot, RTE improves success in most settings with one decay rate. Matched-stride controls attribute most of these gains to aggregation rather than query frequency; RTE recovers 0.67 of the pass@4 headroom on π_0-Long, and its blend contribution on RoboTwin tracks the effective ensemble size.

2 Related Work

Generative Action Policies. Following the success of Diffusion Policy (Chi et al., 2023), generative action representations have become ubiquitous in embodied AI, offering high-precision, continuous multi-step control that discrete autoregressive models struggle to match. CogACT (Li et al., 2024) and RDT (Liu et al., 2025a) integrate diffusion-based action generation into robust vision-language backbones, while the π_0 family (Black et al., 2024; 2025a) and GR00T-N1.7 (Bjorck et al., 2025) leverage flow matching. Regardless of the specific generative mechanism, these models exhibit the same interface: single queries map states and noise to entire action chunks, and RTE operates on these output chunks alone.

Temporal Aggregation in Chunked Control. Querying the policy at intervals shorter than the horizon generates overlapping action predictions. ACT (Zhao et al., 2023) aligns these chunks by absolute temporal target and averages them with exponential weights that favor the oldest prediction, BAKU (Haldar et al., 2024) adopts the same scheme, and Mantis (Yang et al., 2025b) applies it only when instruction-relevant image regions change. CogACT's Adaptive Action Ensemble (AAE) (Li et al., 2024) and VOTE (Lin et al., 2025) instead weight candidates by cosine similarity or trajectory agreement, leaving chunk age unused, while Mixture of Horizons (Jing et al., 2025) trains the policy to fuse chunks of several lengths within a single query. Concurrent work smooths the handoff between consecutive chunks (Zhan et al., 2026; Liu et al., 2026a; Shang et al., 2026), and Lazzati et al. (2026) trace part of the benefit of chunking to implicit ensembling over temporal relationships. RTE retains ACT's alignment and exponential weights, and our analysis casts its decay rate as a trade-off between plan drift and prediction noise whose two limits are both suboptimal.

Test-Time Scaling for Robotics. Current test-time scaling for embodied agents largely focuses on search and selection. BID (Liu et al., 2025b) scores proposals by comparing their coherence to a weaker base policy; RoboMonkey (Kwok et al., 2025) and TACO (Yang et al., 2025a) train verifiers to rank rollouts or actions; E-TTS (Ye et al., 2026) refines reasoning-action pairs using VLM verifiers; and CheckVLA (Liu et al., 2026b) and ELASTIC (Li et al., 2026a) verify execution with a world model or allocate test-time computation per state. These selection-based methods require substantial compute for multiple forward passes or additional learned modules. In contrast, RTE leverages the inherent overlap of standard receding-horizon control, circumventing the need for extra candidates or verifiers. Another line optimizes *when* to query the policy (Nie et al., 2026; Zhao et al., 2026; Feng et al., 2026); since RTE is agnostic to the query schedule, these methods are complementary. Real-Time Chunking (Black et al., 2025b) mitigates inference latency by freezing committed actions and in-painting the rest.

3 Diagnosing Stochastic Headroom

Let x denote the fixed parameters at the start of an episode—including the scene, initial robot state, and language instruction—and let ζ encapsulate the sequence of noise draws generated during the closed-loop rollout. We denote the episode outcome as $Y(x, \zeta) \in \{0, 1\}$.

To decouple these factors, we conduct an episode-level pass@N evaluation. For each fixed initialization, we execute the complete rollout multiple times using the identical checkpoint, instruction, control horizon, and simulator state, varying only the policy noise. Success is defined as:

$$\text{pass@}N = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbf{1}\left[\max_{1 \le n \le N} Y(x, \zeta_n) = 1 \right], \tag{1}$$

where a configuration is considered solved if at least one of the N attempts succeeds.

Table 9 shows that every suite other than the two Long suites largely saturates within four draws. We term the gap pass@N − pass@1 the *stochastic headroom*. A deployed robot, however, can neither replay an episode nor tell in advance which draw will succeed, which raises the question of whether part of this headroom can be recovered within a single rollout.

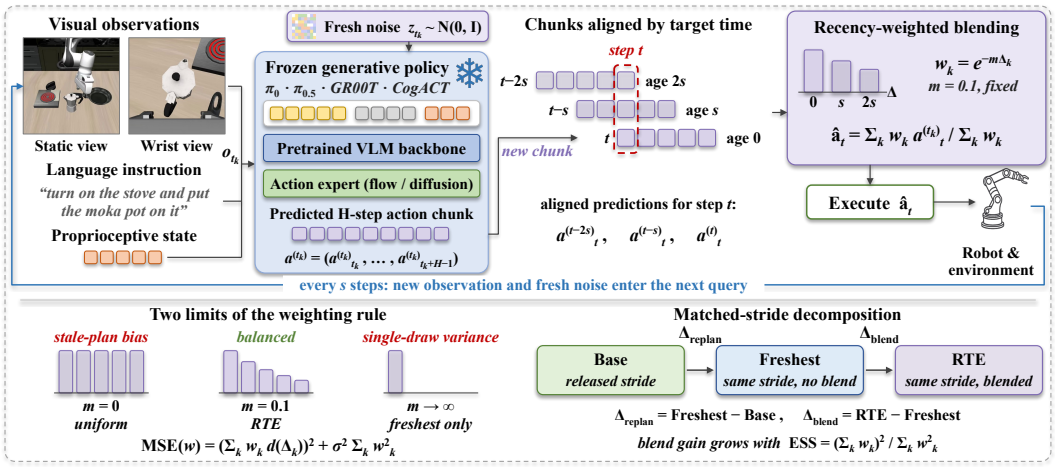


Figure 2: **The RTE framework.** Replanning every s steps produces overlapping action chunks, each conditioned on a new observation and a fresh noise draw. RTE blends the predictions that target the same control step with weights $e^{-m\Delta_k}$, entirely on the client side. *Bottom:* uniform averaging ($m = 0$) incurs stale-plan bias, and freshest-only execution ($m \to \infty$) single-draw variance.

4 Recovering Stochastic Headroom Online

4.1 Plan Drift in Overlapping Chunks

A diffusion or flow-matching policy generates a horizon-H action chunk conditioned on the current observation o_{t_k}, the language instruction ℓ, and a latent noise draw z_{t_k}, which the sampler draws afresh from its random number generator at every query:

$$\boldsymbol{a}^{(t_k)} = G_\theta(o_{t_k}, \ell, z_{t_k}) = \left(a^{(t_k)}_{t_k}, \ldots, a^{(t_k)}_{t_k+H-1}\right), \qquad z_{t_k} \sim p(z). \tag{2}$$

When the policy is queried at a stride $s < H$, the generated chunks overlap, so the action dispatched at time t can have up to $\lceil H/s \rceil$ predictions, one from each chunk generated within the last H steps, each conditioned on a different observation and initialized from different noise.

These samples are non-identically distributed: fresher chunks incorporate realized closed-loop dynamics, whereas older chunks rely on diverging open-loop projections. To quantify this divergence, Figure 1b compares the root mean square (RMS) pose disagreement across prediction age against that of repeated sampling on identical observations, over 1,500 rollouts of π_0 on LIBERO-Long, 500 for each of seeds 7, 8, and 9. While pure sampling noise remains flat, disagreement with the freshest prediction grows with age. Consequently, while historical chunks provide independent noise draws that reduce sampling variance, uniform averaging accumulates stale-plan bias; executing only the freshest prediction incurs the full variance of a single draw.

4.2 Recency-Weighted Temporal Ensembling

As illustrated in Figure 2, RTE addresses this trade-off via a lightweight, client-side buffer. Generated actions are cached alongside their absolute target execution time. At control step t, the system retrieves the valid, active chunks $\mathcal{K}_t$ predicting an action for time t. Given that chunk k was generated at $t_k \le t$, its physical age is $\Delta_{k,t} = t - t_k$. The aggregated execution action is formulated as:

$$\hat{a}_t = \frac{\sum_{k \in \mathcal{K}_t} e^{-m\Delta_{k,t}} a^{(t_k)}_t}{\sum_{k \in \mathcal{K}_t} e^{-m\Delta_{k,t}}}, \qquad m \ge 0. \tag{3}$$

Alignment is based on target time rather than intra-chunk index. The same temporal weight is applied across all action coordinates. The limits of m correspond to the matched controls of Section 5.1: $m = 0$ yields uniform averaging, while $m \to \infty$ reverts to freshest-only execution. Algorithm 1 in Appendix B outlines the control step; this client-side blending introduces negligible computational overhead relative to policy inference.

4.3 DRIFT-VARIANCE ANALYSIS

To analyze how the weights should depend on prediction age, we model the aligned prediction of a single action coordinate at step t from a chunk of age Δ_k as

$$a_t^{(t_k)} = a_t^\star + d(\Delta_k) + \varepsilon_k, \tag{4}$$

where $a_t^\star$ is the ideal action, $d(\cdot) \geq 0$ is a non-decreasing drift with $d(0)=0$, and ε_k is zero-mean noise with variance σ^2, independent across chunks. As each chunk has its own observation and noise draw, ε_k covers both sampling noise and the policy's error on that observation, which we jointly call prediction noise. For weights with $\sum_k w_k=1$, the mean squared error (MSE) splits into stale-plan bias and noise variance, $\mathrm{MSE}(w) = \big(\sum_k w_k d(\Delta_k)\big)^2 + \sigma^2 \sum_k w_k^2$, so spreading weight over older chunks lowers variance but raises bias. Since actions are bounded, we model the drift as saturating:

Assumption 1 (Saturating drift)**.** $d(\Delta) = d_\infty\big(1 - e^{-m_0\Delta}\big)$ for some asymptote $d_\infty > 0$ and drift rate $m_0 > 0$.

Proposition 1 (Optimal weights)**.** *Let $n \geq 2$ chunks be active and $\sigma^2 > 0$. Under Assumption 1, the weights minimizing $\mathrm{MSE}(w)$ subject to $\sum_k w_k=1$ are $w_k^\star = \alpha + \beta\, e^{-m_0\Delta_k}$ with $\beta > 0$. They decrease strictly with age, and their age-dependent part decays at the drift rate m_0.*

As derived in Appendix A, the constrained optimum is $w_j^\star = (\mu - B\, d(\Delta_j))/\sigma^2$ with constants $\mu > 0$ and $B \geq 0$, so the optimal weights mirror the drift profile whatever its shape, and Assumption 1 turns this mirror image into an exponential in age. The exact solution also carries a uniform offset α, positive when noise dominates and negative when drift dominates, in which case the stalest chunks can receive negative weight. RTE is therefore a one-parameter approximation rather than the exact minimizer: it keeps the exponential component and drops the offset, $w_k \propto e^{-m\Delta_k}$, which keeps every weight positive and leaves a single rate m, not tied to m_0, to trade bias against variance. If the stale-plan error instead accumulates the plan innovations since each query, the offset vanishes and, apart from an end-point term, the optimum decays exponentially in age, the classical optimality of exponential smoothing (Muth, 1960), as Appendix A.3 shows.

Neither end of this family is optimal at a replanning instant, whatever the form of the drift. For any non-decreasing, non-constant drift, $\mathrm{MSE}'(0) = -2\bar{d}\,\mathrm{Cov}(\Delta, d) < 0$ over the active chunks, which rules out uniform averaging, and the error approaches its freshest-only value σ^2 from below as $m \to \infty$, so the optimal rate is finite and positive, as Proposition 2 in Appendix A shows. Furthermore, because the variance term depends on the weights only through $\sum_k w_k^2$, the effective ensemble size $\mathrm{ESS} = (\sum_k w_k)^2 / \sum_k w_k^2$, Kish's effective sample size (Kish, 1965), measures how much averaging a weighting delivers; Figure 4c relates it to the empirical gain of blending.

5 EXPERIMENTS

Our experiments are designed to answer three questions. (1) Does RTE recover the stochastic headroom online across backbones, benchmarks, and physical robots? (2) How much of its gain comes from temporal aggregation rather than from faster replanning? (3) How should the overlapping predictions be weighted, and when does aggregation help?

5.1 EXPERIMENTAL SETUP

Simulation Benchmarks and Baselines. We evaluate on three simulation benchmarks, LIBERO (Liu et al., 2023), SIMPLER (Li et al., 2025), and the 50-task RoboTwin 2.0 (Chen et al., 2025b; Mu et al., 2025), with four chunked generative policies: the diffusion-based CogACT (Li et al., 2024), evaluated on LIBERO and SIMPLER, and the flow-matching π_0, $\pi_{0.5}$ (Black et al., 2024; 2025a), and GR00T-N1.7 (Bjorck et al., 2025), evaluated on LIBERO and RoboTwin. Wherever a public checkpoint targets the benchmark, we evaluate it directly, and otherwise we fine-tune from public bases, as detailed in Appendix B.

Evaluation Settings. The decay rate is selected once on π_0-Long (seed 7) and frozen at $m = 0.1$ across all backbones, benchmarks, and seeds. On LIBERO (500 episodes per suite across seeds 7/8/9), models are evaluated at two strides: $s \in \{3, 5\}$ for the π_0 family and $s \in \{4, 8\}$ for GR00T and CogACT (for CogACT, whose release commits 12–16 actions per chunk with minimal overlap,

Table 1: **LIBERO success rates (%) at two strides**, mean ± population std over seeds $7/8/9$ (500 episodes per suite and seed). Freshest executes only the newest chunk and Equal averages the aligned chunks uniformly; both share the query schedule of RTE at each stride, and Δ is RTE minus Freshest. RTE is shaded, and bold marks the best of the three rules.

Model	Stride	Method	Spatial	Object	Goal	Long	Average
CogACT	s=8	Freshest	93.3 ± 0.5	$\mathbf{96.4 \pm 0.4}$	91.6 ± 1.2	82.3 ± 0.5	90.9 ± 0.5
		Equal	93.3 ± 0.4	94.0 ± 0.3	$\mathbf{93.3 \pm 0.7}$	$\mathbf{90.4 \pm 0.7}$	92.7 ± 0.4
		RTE	$\mathbf{93.8 \pm 0.7}$	95.8 ± 0.5	92.9 ± 1.0	89.7 ± 0.5	$\mathbf{93.1 \pm 0.4}$
		Δ	$+0.5$	-0.6	$+1.3$	$+7.4$	$+2.2$
	s=4	Freshest	87.1 ± 1.4	91.6 ± 0.6	83.1 ± 0.8	65.9 ± 0.9	81.9 ± 0.8
		Equal	93.4 ± 0.7	$\mathbf{97.3 \pm 0.7}$	$\mathbf{92.9 \pm 1.1}$	$\mathbf{90.0 \pm 1.0}$	$\mathbf{93.4 \pm 0.6}$
		RTE	$\mathbf{94.3 \pm 1.0}$	96.7 ± 0.4	91.5 ± 1.1	87.4 ± 0.9	92.5 ± 0.6
		Δ	$+7.1$	$+5.1$	$+8.5$	$+21.5$	$+10.6$
π_0	s=5	Freshest	96.9 ± 0.4	97.3 ± 0.4	93.7 ± 0.9	80.7 ± 0.5	92.2 ± 0.2
		Equal	92.7 ± 0.6	95.3 ± 0.4	93.0 ± 0.7	80.9 ± 0.6	90.5 ± 0.3
		RTE	$\mathbf{97.7 \pm 1.1}$	$\mathbf{99.3 \pm 0.6}$	$\mathbf{96.5 \pm 0.6}$	$\mathbf{90.1 \pm 0.5}$	$\mathbf{95.9 \pm 0.3}$
		Δ	$+0.7$	$+2.0$	$+2.8$	$+9.4$	$+3.7$
	s=3	Freshest	96.3 ± 0.5	96.5 ± 0.7	90.5 ± 1.0	77.9 ± 1.9	90.3 ± 0.6
		Equal	94.1 ± 1.1	95.8 ± 0.8	93.1 ± 0.6	81.9 ± 1.9	91.2 ± 0.6
		RTE	$\mathbf{97.9 \pm 0.2}$	$\mathbf{98.7 \pm 0.5}$	$\mathbf{97.0 \pm 0.3}$	$\mathbf{89.9 \pm 0.9}$	$\mathbf{95.9 \pm 0.4}$
		Δ	$+1.6$	$+2.1$	$+6.5$	$+12.0$	$+5.6$
$\pi_{0.5}$	s=5	Freshest	98.2 ± 0.3	98.5 ± 0.4	97.9 ± 0.6	92.5 ± 0.7	96.8 ± 0.1
		Equal	98.7 ± 0.8	$\mathbf{99.0 \pm 0.5}$	$\mathbf{98.3 \pm 0.6}$	$\mathbf{94.9 \pm 0.6}$	$\mathbf{97.7 \pm 0.5}$
		RTE	$\mathbf{98.9 \pm 0.2}$	95.5 ± 1.1	98.1 ± 0.5	94.3 ± 0.5	96.7 ± 0.1
		Δ	$+0.7$	-3.0	$+0.2$	$+1.8$	-0.1
	s=3	Freshest	98.1 ± 0.8	97.3 ± 1.0	96.5 ± 0.8	91.3 ± 0.6	95.8 ± 0.7
		Equal	$\mathbf{98.3 \pm 0.2}$	98.9 ± 0.4	$\mathbf{98.1 \pm 0.2}$	93.5 ± 0.9	97.2 ± 0.4
		RTE	$\mathbf{98.3 \pm 0.8}$	$\mathbf{99.5 \pm 0.2}$	97.9 ± 0.4	$\mathbf{94.5 \pm 0.7}$	$\mathbf{97.5 \pm 0.5}$
		Δ	$+0.2$	$+2.2$	$+1.5$	$+3.1$	$+1.8$
GR00T-N1.7	s=8	Freshest	97.7 ± 0.3	99.1 ± 0.2	97.7 ± 0.7	92.1 ± 1.0	96.7 ± 0.4
		Equal	$\mathbf{100.0 \pm 0.0}$	$\mathbf{99.6 \pm 0.3}$	$\mathbf{99.5 \pm 0.1}$	92.2 ± 0.6	$\mathbf{97.8 \pm 0.1}$
		RTE	97.2 ± 1.1	99.5 ± 0.3	98.3 ± 0.1	$\mathbf{93.2 \pm 1.3}$	97.1 ± 0.6
		Δ	-0.5	$+0.4$	$+0.5$	$+1.1$	$+0.4$
	s=4	Freshest	98.5 ± 1.0	99.3 ± 0.4	97.7 ± 0.8	90.3 ± 1.6	96.5 ± 0.9
		Equal	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{99.6 \pm 0.2}$	$\mathbf{92.3 \pm 0.1}$	$\mathbf{98.0 \pm 0.0}$
		RTE	97.1 ± 0.4	99.8 ± 0.2	97.3 ± 0.7	92.1 ± 1.2	96.6 ± 0.1
		Δ	-1.4	$+0.5$	-0.4	$+1.8$	$+0.1$

s=8 enables blending). On RoboTwin 2.0 (100 rollouts per task, `demo_clean`), the open-loop Base executes full chunks, while RTE replans at $s \in \{10, 25\}$ (H=50) and $s \in \{4, 8\}$ (H=16). All runs use NVIDIA A100 GPUs with identical initial states per comparison. We compare RTE against matched-stride controls: *Freshest* (executing only the newest chunk) and *Equal* (uniform averaging).

5.2 Simulation Results

LIBERO. As shown in Table 1, RTE achieves its largest gains where stochastic headroom remains substantial. On π_0-Long, it boosts success from 80.7% to 90.1%, closing 0.67 of the gap to pass@4, and its margin widens under faster replanning (s=3) where unaggregated query noise degrades the freshest baseline. Indeed, frequent replanning without aggregation exposes policies to compounding sampling variance: for CogACT, its Long success falls to 65.9% at s=4, and blending restores it to 87.4%, although neither stride reaches its release, which commits to longer open-loop segments (reaching 94.6%, Table 7). Uniform averaging (m=0) exhibits horizon-dependent behavior: for the 50-step chunks of π_0, accumulating stale-plan bias drops it below the freshest baseline on Spatial and Object, whereas for short horizons ($H \leq 16$), predictions remain sufficiently fresh that it is on par with RTE and ahead in four of the six settings. Paired episode attribution on π_0-Long (Table 10) shows that RTE mainly recovers stochastic failures: as traced in Figure 1c, it resolves 79.7% of the Base failures that resampling can fix, but only 42.3% of the remainder. A fresh rollout of Base recovers far fewer of these failures and loses more of its successes.

SIMPLER. CogACT natively ensembles actions with Adaptive Action Ensemble (AAE), which weights predictions by their cosine similarity to the newest one. Table 2 isolates this weighting

Table 2: **CogACT on the four SIMPLER Google Robot Visual Matching tasks**, success rates (%), mean ± population std over seeds 7/8/9. All four rules query at every step with the released dual-candidate buffer and share the checkpoint; Freshest executes only the newest prediction and Equal averages the two candidates uniformly. The RTE row is shaded, bold marks the best of the four rules, and Δ is RTE minus AAE.

Method	Pick Coke Can	Move Near	Open/Close Drawer	Open Top Drawer and Place Apple	Average
CogACT (AAE)	92.0 ± 1.4	83.9 ± 1.4	71.3 ± 0.8	45.7 ± 2.4	73.2 ± 0.4
CogACT (Freshest)	91.3 ± 0.5	$\mathbf{86.1 \pm 1.4}$	$\mathbf{74.5 \pm 1.1}$	44.8 ± 3.9	74.2 ± 1.0
CogACT (Equal)	92.8 ± 1.1	86.0 ± 0.7	72.1 ± 0.4	38.6 ± 2.4	72.3 ± 1.1
CogACT (RTE)	$\mathbf{94.3 \pm 1.0}$	84.9 ± 0.9	73.6 ± 0.4	$\mathbf{47.5 \pm 3.1}$	$\mathbf{75.1 \pm 0.7}$
Δ	+2.3	+1.0	+2.3	+1.9	+1.9

Table 3: **Matched-stride decomposition on RoboTwin 2.0**, success rates (%) over 50 tasks (`demo_clean`, 100 rollouts per task). Freshest, Equal, and RTE share the query schedule at each stride; $\Delta_{\text{replan}} = \text{Freshest} - \text{Base}$, $\Delta_{\text{blend}} = \text{RTE} - \text{Freshest}$, and $\Delta = \text{RTE} - \text{Base}$; ESS is that of the RTE weights. RTE is shaded, and bold marks the best rule per row.

Model	Stride	ESS	Base	Freshest	Equal	RTE	Δ_{replan}	Δ_{blend}	Δ
π_0 (H=50)	s=25	1.2	33.9	**43.4**	39.4	42.9	+9.5	−0.5	+8.9
	s=10	2.1		37.0	**43.5**	43.4	+3.0	+6.4	+9.5
$\pi_{0.5}$ (H=50)	s=25	1.2	57.7	71.7	68.6	**72.3**	+14.0	+0.6	+14.6
	s=10	2.1		73.8	74.3	**78.7**	+16.1	+4.8	+20.9
GR00T-N1.7 (H=16)	s=8	1.7	32.7	30.5	**35.8**	34.8	−2.2	+4.3	+2.1
	s=4	3.4		26.2	**35.2**	34.4	−6.5	+8.2	+1.7

function on CogACT's released SIMPLER setup. Replacing similarity with prediction age improves all four Visual Matching tasks and raises the average success rate from 73.2% to 75.1%. With two candidates one step apart, similarity weighting at α=0.1 stays close to uniform, whereas age weighting consistently tilts toward the newer prediction. Both limits of the family fall short of this tilted blend on average, and uniform averaging, which AAE approximates, is the weakest of the four rules.

RoboTwin 2.0. Table 3 separates each gain over the open-loop Base into faster replanning, Δ_{replan}, and blending at the same stride, Δ_{blend}. Replanning alone helps the H=50 policies but not monotonically, as the freshest-only control of π_0 loses ground from s=25 to s=10. Blending cancels this loss for π_0 and turns the extra queries into a gain for $\pi_{0.5}$, which rises from 57.7% to 78.7%, whereas uniform averaging reaches only 74.3%. At s=25, where the buffer holds little more than one effective chunk, blending barely changes the averages, while uniform averaging, which gives the older chunk half the weight, costs π_0 four points against the freshest-only control. GR00T, with H=16, loses ground when it replans without aggregation, and blending recovers the loss; as on LIBERO, uniform averaging does slightly better on its short chunks. Across the three backbones, the blend contribution tracks the effective ensemble size listed in Table 3: it stays near zero at an ESS of 1.2 and turns positive from 1.7 upward, as Figure 4c shows.

5.3 Real-World Evaluation

Real-World Setup. To validate RTE under unmodeled physical dynamics, we deploy our framework on a 7-DoF Franka Research 3 arm equipped with a parallel-jaw gripper and a static top-view ZED 2i camera, and evaluate the five tasks shown in Figure 3. We fine-tune each of the four backbones (CogACT, π_0, $\pi_{0.5}$, and GR00T-N1.7) on all five tasks jointly from 50 expert demonstrations per task. Each backbone is deployed in three ways: Base executes each predicted chunk open-loop, while the freshest-only control and RTE replan at the same stride, $s = 10$ for π_0, $\pi_{0.5}$, and GR00T-N1.7 and $s = 8$ for CogACT. Each configuration is evaluated over 40 rollouts per task on layouts shared by the three deployments; task definitions are given in Appendix E, and training configurations in Table 6.

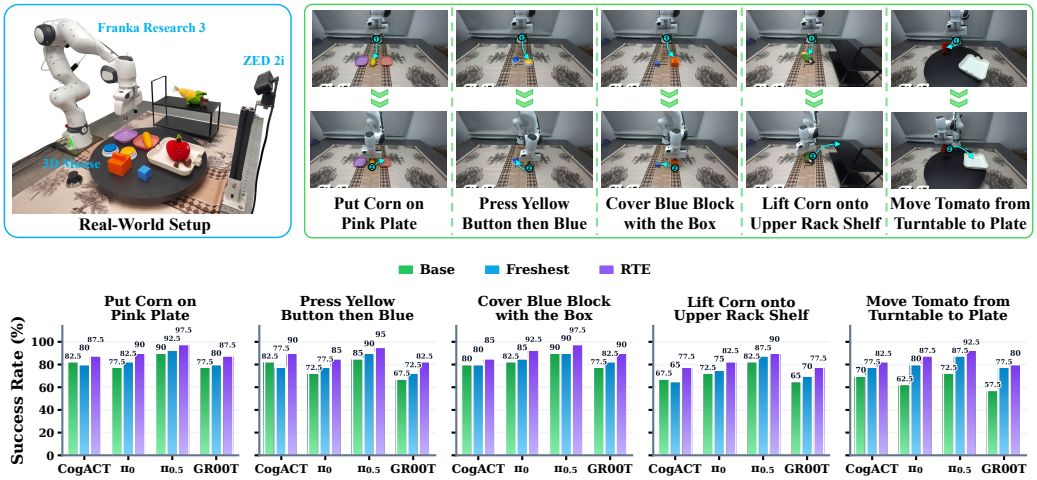


Figure 3: **Real-world setup and results.** *Top*: the Franka Research 3 arm with a static top-view ZED 2i camera, and the five tasks with numbered waypoints. *Bottom*: per-task success rates over 40 rollouts per cell.

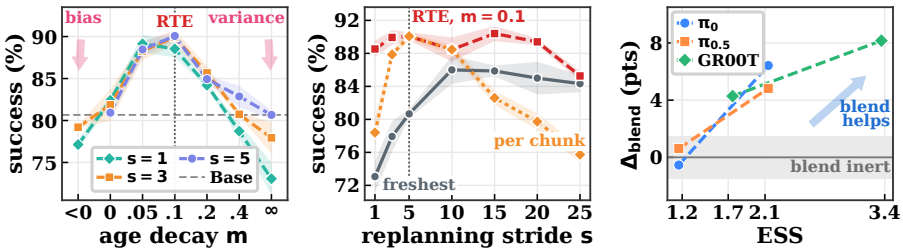
Table 4: **Estimator ablation on π_0-Long**, success rates (%), mean $\pm$ population std over seeds 7/8/9; Δ is measured against Base in the first row. *Replan* is the query stride, *Blend* marks temporal aggregation, *Recency* marks weights that favor newer predictions, and bold marks the best weighting rule at s=3.

Configuration	Replan	Blend	Recency	Success Rate (%)	Δ
Base (released stride)	s=5	×	–	80.7 ± 0.5	—
Without temporal aggregation					
Open loop (full chunk)	s=50	×	–	66.4 ± 1.9	−14.3
Freshest only	s=3	×	–	77.9 ± 1.9	−2.7
Freshest only	s=1	×	–	73.1 ± 1.9	−7.6
Four samples per query	s=5	×	–	80.3 ± 1.0	−0.4
Temporal aggregation at matched query schedules					
RTE (m=0.1)	s=5	✓	✓	90.1 ± 0.5	+9.4
RTE (m=0.1)	s=1	✓	✓	88.5 ± 1.0	+7.9
Weighting rule					
Equal weights (m=0)	s=3	✓	×	81.9 ± 1.9	+1.2
ACT (oldest first)	s=3	✓	×	79.2 ± 0.2	−1.5
AAE (similarity)	s=3	✓	×	81.1 ± 2.0	+0.4
VOTE	s=3	✓	×	83.9 ± 1.9	+3.3
AAE + age	s=3	✓	✓	88.9 ± 0.8	+8.3
RTE (m=0.1)	s=3	✓	✓	**89.9 ± 0.9**	**+9.3**

Results. As illustrated in the lower panel of Figure 3, RTE improves success on all sixteen backbone–task pairs of the first four tasks over both open-loop Base and the freshest-only control, and most of this gain comes from blending rather than from faster replanning: averaged over these tasks, RTE raises π_0 from 76.3% to 87.5% and GR00T-N1.7 from 71.9% to 84.4%, while CogACT again loses ground when it replans without aggregation and more than recovers it once the chunks are blended. The turntable task reverses this balance: its target keeps moving, so faster replanning brings the largest gain for every backbone, while blending adds a consistent further improvement upon the freshest-only control.

5.4 Ablation Studies

Replanning Frequency vs. Temporal Aggregation. Evaluating matched query schedules in Table 4 shows that replanning without aggregation systematically degrades performance—falling down to



(a) Decay rate (b) Replanning stride (c) Blending vs. ESS

Figure 4: **Decay rate, stride, and effective ensemble size.** (a) Success on π_0-Long along the decay rate, from ACT's oldest-first rule ($m{<}0$) to the freshest chunk alone ($m{\to}\infty$). (b) Success on the same suite along the stride for the freshest-only control, RTE, and a per-chunk decay matched to $m{=}0.1$ at $s{=}5$; dotted: released stride. Bands: three-seed std. (c) Δ_{blend} of Table 3 against the ESS on RoboTwin 2.0; grey band: $|\Delta_{\text{blend}}| < 1.5$ points.

73.1% at $s{=}1$ despite five times the queries. In contrast, RTE stays near 90% at every stride and gains 9.4 points even at the released stride, verifying that the improvement on LIBERO-Long originates from temporal blending rather than query frequency. Averaging four fresh samples per query reaches only 80.3%: as formalized in Appendix A.4, draws conditioned on the same observation share the policy's error on that observation, whereas overlapping chunks are conditioned on different observations, so their errors partly cancel. Trajectory smoothness alone cannot explain these gains: uniform averaging produces smoother motions without improving success, and low-pass filtering freshest-only actions degrades performance (Table 17).

The Weighting Rule and Decay Rate. As predicted by Proposition 2, both limits of Equation (3) underperform interior decay rates at every stride in Figure 4a. Existing ensembling heuristics also lag behind age weighting on identical checkpoints: ACT's oldest-first weights, CogACT's similarity weights, and VOTE's majority filter all stay below 84%, whereas modulating similarity by the age factor of RTE recovers 88.9%, isolating prediction age as the critical signal. Crucially, the advantage of recency over uniform weighting grows with prediction age rather than chunk count: in Figure 6, it widens on π_0-Long from $s{=}2$, where twenty-five chunks overlap, to $s{=}25$, where uniform averaging allocates half its weight to a 25-step-stale plan. Across the four backbones of that sweep, the best rate for π_0 remains within $[0.05, 0.2]$ up to $s{=}40$, while uniform weights only approach the optimum when predictions remain young ($\pi_{0.5}$) or single draws are noisy (CogACT). The rate $m{=}0.1$ is thus a common operating point rather than the optimum of every backbone.

Replanning Stride. Extending the comparison to longer strides (Figure 4b) shows that committing longer to each plan cannot match temporal ensembling: even the best freshest-only stride falls 4.1 points short of RTE at the released stride, while RTE holds 88–90% up to $s{=}20$ with a quarter of the queries. Parameterizing decay per step is essential: chunk-fixed decays hold only near their calibrated stride, and slower per-step decays degrade at long strides. The rate thus acts as a time constant: a prediction's weight halves every $\ln 2/m \approx 7$ control steps whatever the stride, so m sets how far back in time the blend reaches, and shorter strides mainly add chunks conditioned on nearly the same observations, which likely share much of their error.

6 Conclusion

We introduce *stochastic headroom*, an episode-level pass@N measure of how much of a generative policy's failure is exposed by resampling alone, and show that RTE, a training-free estimator built on the overlap inherent to receding-horizon control, recovers much of this headroom online. Weighting aligned predictions by their age averages out prediction noise while limiting stale-plan bias, adds no queries at a given stride, and improves most tested settings across backbones and benchmarks with one decay constant. Matched-stride controls attribute these gains primarily to temporal aggregation rather than query frequency, a benefit that faster replanning alone cannot achieve, and show that the blend contribution tracks the effective ensemble size. We discuss the operational scope, limitations, and future directions in Appendix F.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of proofs. Additionally, we used generative AI tools to polish the writing of this paper to improve readability, to help draft parts of the text, to identify and summarize related literature, and to assist with writing code. We have reviewed all AI-assisted work. We verified every AI-assisted derivation, reviewed and edited all AI-assisted text, and reviewed and tested all AI-assisted code. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies an inference-time method for making generative robot manipulation policies more reliable. All experiments ran in simulation (LIBERO, SIMPLER, RoboTwin 2.0) or on a tabletop Franka Research 3 arm under human supervision with software and hardware emergency stops. The study involves no human subjects, personal data, or hazardous materials. More reliable manipulation can help industrial automation and assistive care; like other advances in autonomous manipulation, it also carries dual-use risk that deployment guidelines should address.

REPRODUCIBILITY STATEMENT

Section 5.1 and Appendix B specify the benchmarks, checkpoints, evaluation seeds, episode counts, and deployment settings of every experiment, and Appendix B also lists the training configurations and hardware. Appendix A states the assumptions and gives complete proofs for the claims in Section 4.3. Code for RTE and the evaluation wrappers will be released upon publication.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *CoRR*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL `https://arxiv.org/abs/2410.24164`.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Annual Conference on Robot Learning*, 2025a.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

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

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Xiang Li, Quanlu Zhang, Zhaofei Yu, Guoliang Fan, Tiejun Huang, Yu Wang, and Chao Yu. $\pi_{\texttt{rl}}$: Online RL fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 2025a.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, 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*, 2025b.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems XIX*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Xiangdong Feng, Yuxuan Cheng, Chen Shi, Boyao Han, Yuxuan Yan, Yitong Hong, Zhuotao Tian, and Li Jiang. Denoising tells when to replan: Denoising-variance adaptive chunking for flow-based robot policies. *arXiv preprint arXiv:2606.03847*, 2026.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024.

Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. BAKU: An efficient transformer for multi-task policy learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6840–6851, 2020.

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

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning*, pp. 2679–2713. PMLR, 2025b.

Leslie Kish. *Survey Sampling*. John Wiley & Sons, 1965.

Hadas Kress-Gazit, Kunimatsu Hashimoto, Naveen Kuppuswamy, Paarth Shah, Phoebe Horgan, Gordon Richardson, Siyuan Feng, and Benjamin Burchfiel. Robot learning as an empirical science: Best practices for policy evaluation. *arXiv preprint arXiv:2409.09491*, 2024.

Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, and Percy Liang. SPoC: Search-based pseudocode to code. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. RoboMonkey: Scaling test-time sampling and verification for vision-language-action models. In *Conference on Robot Learning (CoRL)*, 2025.

Filippo Lazzati, Kyle Stachowicz, William Chen, Alberto Maria Metelli, Andrew Wagenmaker, and Sergey Levine. Why does action chunking improve behavioral cloning performance in robotic control? *arXiv preprint arXiv:2608.02547*, 2026.

Andrew Zou Li, Gokul Swamy, Yonatan Bisk, and Andrea Bajcsy. ELASTIC: Efficiently learning to adaptively scale test-time compute for generative control policies. *arXiv preprint arXiv:2606.31132*, 2026a.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, Dehui Wang, Dingxiang Luo, Yuchen Fan, Youbang Sun, Jia Zeng, Jiangmiao Pang, Shanghang Zhang, Yu Wang, Yao Mu, Bowen Zhou, and Ning Ding. SimpleVLA-RL: Scaling VLA training via reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026b.

Peiyan Li, Yuze Zhu, Yixiang Chen, Qisen Ma, Yuan Xu, Jiabing Yang, He Guan, Yan Huang, Hongtao Wu, Xiao Ma, et al. BridgeVLA++: A data-efficient, generalizable, and memory-augmented vision-language-action framework for 3d manipulation. *arXiv preprint arXiv:2608.05042*, 2026c.

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

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning*, pp. 3705–3728. PMLR, 2025.

Juyi Lin, Amir Taherin, Arash Akbari, Arman Akbari, Lei Lu, Guangyu Chen, Taskin Padir, Xiaomeng Yang, Weiwei Chen, Yiqian Li, et al. VOTE: vision-language-action optimization with trajectory ensemble voting. *arXiv preprint arXiv:2507.05116*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

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

Dongyang Liu, Zhaowen Zheng, Yu Sun, Longxu Zhang, Yixuan Liu, and Hao Wan. Action-prior denoising for smooth real-time chunking. *arXiv preprint arXiv:2605.25537*, 2026a.

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 *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025a.

Yuejiang Liu, Jubayer Ibn Hamid, Annie Xie, Yoonho Lee, Maximilian Du, and Chelsea Finn. Bidirectional Decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations (ICLR)*, 2025b.

Yushan Liu, Peibo Sun, Xintao Chao, Zhenyang Yang, Yifan Xie, Lingfeng Zhang, Shoujie Li, Chenyu Tang, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. CheckVLA: Execution-time verification with action-conditioned world model for long-horizon mobile manipulation. *arXiv preprint arXiv:2607.26789*, 2026b.

David Q. Mayne and Hannah Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, 1990.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, et al. RoboTwin: Dual-arm robot benchmark with generative digital twins. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 27649–27660, 2025.

John F. Muth. Optimal properties of exponentially weighted forecasts. *Journal of the American Statistical Association*, 55(290):299–306, 1960.

Junnan Nie, Jiayi Li, Jiachen Zhang, Junyi Lao, Chenghao Liu, Tianle Zhang, and Songfang Huang. PACE: Phase-aware chunk execution for robot policies with action chunking. *arXiv preprint arXiv:2606.00537*, 2026.

Ming Shang, Yuchen Huang, Jiaoyang Chen, Haoyuan Hu, Han Yu, Liping Song, Luyun Feng, Shuo Bao, Wei Dong, Xinzhou Wang, and Fuchun Sun. BICPO-VLA: Behavior-identified continuation preference optimization for smooth asynchronous vision-language-action control. *arXiv preprint arXiv:2608.13924*, 2026.

Joseph A Vincent, Haruki Nishimura, Masha Itkina, Paarth Shah, Mac Schwager, and Thomas Kollar. How generalizable is my behavior cloning policy? a statistical approach to trustworthy performance evaluation. *IEEE Robotics and Automation Letters*, 9(10):8619–8626, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Siyuan Yang, Yang Zhang, Haoran He, Ling Pan, Xiu Li, Chenjia Bai, and Xuelong Li. Steering vision-language-action models as anti-exploration: A test-time scaling approach. *arXiv preprint arXiv:2512.02834*, 2025a.

Yi Yang et al. Mantis: A versatile vision-language-action model with disentangled visual foresight. *arXiv preprint arXiv:2511.16175*, 2025b.

Wen Ye, Peiyan Li, Tingyu Yuan, Yuan Xu, Xiangnan Wu, Chaoyang Zhao, Jing Liu, Nianfeng Liu, Yan Huang, and Liang Wang. E-TTS: A new embodied test-time scaling framework for robotic manipulation. *arXiv preprint arXiv:2606.27268*, 2026.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Dijia Zhan, Xuemiao Xu, Jinyi Li, and Jie Tang. SEAM: Smooth execution of action-chunked motion for vision-language-action policies. *arXiv preprint arXiv:2607.04609*, 2026.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

Yuchi Zhao, Miroslav Bogdanovic, Arjun Sohal, Liyu Tao, Kourosh Darvish, Alán Aspuru-Guzik, Florian Shkurti, and Animesh Garg. Dynamic execution horizon prediction for chunk-based robot policies. *arXiv preprint arXiv:2606.11408*, 2026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

APPENDIX CONTENTS

A Theory Derivations **14**

A.1 Constrained Minimizer 15

A.2 The Exponential Family 16

A.3 Nested Plan Innovations 16

A.4 Correlated and Age-Dependent Noise 17

A.5 Measuring Drift and Sampling Noise 18

B Experimental Details **18**

B.1 Implementation 18

B.2 Model Architectures and Checkpoints 18

B.3 Per-Backbone Deployment Details 19

B.4 The Cost of Aggregation 19

C Additional Experimental Results **20**

C.1 Diagnosing Stochastic Headroom 21

C.2 Simulation Results: RoboTwin 2.0 21

C.3 Ablation Studies 22

D Task Visualizations **27**

E Real-World Setup and Tasks **27**

E.1 Hardware and Environment 27

E.2 Task Definitions 28

E.3 Deployment Protocol and Control Deadlines 28

F Limitations and Future Scope **29**

A THEORY DERIVATIONS

This appendix states the model behind Section 4.3 precisely and proves its claims. Appendix A.1 derives the error-minimizing weights and proves Proposition 1, Appendix A.2 analyzes the exponential family used by RTE, Appendix A.3 treats a stale-plan error that accumulates plan innovations, Appendix A.4 relaxes the noise assumptions, and Appendix A.5 describes the measurement behind Figure 1b.

Setting. Fix one action coordinate and one control step t, and let $a^\star$ denote the ideal action at t. Suppose that $n \geq 2$ active chunks predict this step, indexed by increasing age $\Delta_0 < \Delta_1 < \cdots < \Delta_{n-1}$. With a fixed stride s, $\Delta_k = \tau + ks$, where $\tau \in \{0, \ldots, s-1\}$ counts the steps since the last replanning instant. Following Equation (4), chunk k predicts

$$a_k = a^\star + d_k + \varepsilon_k, \qquad d_k := d(\Delta_k),$$

where $d : [0, \infty) \to [0, \infty)$ is non-decreasing with $d(0) = 0$, and $\varepsilon_0, \ldots, \varepsilon_{n-1}$ are uncorrelated with zero mean and common variance $\sigma^2 > 0$. For weights $\boldsymbol{w} = (w_0, \ldots, w_{n-1})^\top$ with $\mathbf{1}^\top \boldsymbol{w} = 1$, the aggregate $\hat{a} = \sum_k w_k a_k$ satisfies $\hat{a} - a^\star = B(\boldsymbol{w}) + N(\boldsymbol{w})$, with stale-plan bias $B(\boldsymbol{w}) := \sum_k w_k d_k$

and aggregated noise $N(\boldsymbol{w}) := \sum_k w_k \varepsilon_k$. Because $\mathbb{E}[N(\boldsymbol{w})] = 0$ and $\mathbb{E}[N(\boldsymbol{w})^2] = \sigma^2 \|\boldsymbol{w}\|^2$,

$$\mathrm{MSE}(\boldsymbol{w}) := \mathbb{E}\big[(\hat{a} - a^\star)^2\big] = B(\boldsymbol{w})^2 + \sigma^2 \|\boldsymbol{w}\|^2. \tag{5}$$

The results of Appendix A.1 hold for any distinct ages; only parts (ii) and (iii) of Proposition 2 and Proposition 3 use the equal spacing. RTE applies one weight vector to every coordinate. If the drift profiles of the coordinates are proportional, $d^{(c)} = \gamma_c \, d$, the error summed over coordinates equals $(\sum_c \gamma_c^2) B(\boldsymbol{w})^2 + (\sum_c \sigma_c^2) \|\boldsymbol{w}\|^2$, which has the form of Equation (5) with rescaled constants, so the results below apply to the full action as well.

A.1 Constrained Minimizer

Lemma 1. *Let $n \geq 2$ and $\sigma^2 > 0$. Over all $\boldsymbol{w} \in \mathbb{R}^n$ with $\mathbf{1}^\top \boldsymbol{w} = 1$, Equation (5) has a unique minimizer $\boldsymbol{w}^\star$, given by*

$$w_j^\star = \frac{\mu - B^\star d_j}{\sigma^2}, \qquad B^\star = \frac{\mu \sum_k d_k}{\sigma^2 + \sum_k d_k^2}, \qquad \mu = \sigma^2 \left[n - \frac{\big(\sum_k d_k\big)^2}{\sigma^2 + \sum_k d_k^2} \right]^{-1}, \tag{6}$$

where $B^\star = B(\boldsymbol{w}^\star)$. Moreover, $\mu > 0$ and $B^\star \geq 0$, with $B^\star > 0$ unless every d_k vanishes. Hence $w_j^\star$ is non-increasing in age, strictly decreasing wherever d increases, and equal to $1/n$ when d is constant on the active ages.

Proof. Let $\boldsymbol{d} = (d_0, \ldots, d_{n-1})^\top$. The objective is $\boldsymbol{w}^\top Q \boldsymbol{w}$ with $Q = \boldsymbol{d}\boldsymbol{d}^\top + \sigma^2 I$, which is positive definite because $\sigma^2 > 0$. A strictly convex quadratic has a unique minimizer on the affine set $\{\boldsymbol{w} : \mathbf{1}^\top \boldsymbol{w} = 1\}$, characterized by the Lagrange condition $Q\boldsymbol{w} = \mu \mathbf{1}$ for some $\mu \in \mathbb{R}$. Its jth component reads $B d_j + \sigma^2 w_j = \mu$ with $B = \boldsymbol{d}^\top \boldsymbol{w}$, which is the first identity. Multiplying it by d_j and summing over j gives $B\big(\sigma^2 + \sum_j d_j^2\big) = \mu \sum_j d_j$, the second. Summing it over j and using $\mathbf{1}^\top \boldsymbol{w} = 1$ gives $n\mu - B \sum_j d_j = \sigma^2$, and substituting B yields the third. By the Cauchy–Schwarz inequality, $\big(\sum_j d_j\big)^2 \leq n \sum_j d_j^2 < n\big(\sigma^2 + \sum_j d_j^2\big)$, so the bracket is positive and $\mu > 0$. Since every $d_j \geq 0$, the second identity gives $B^\star \geq 0$, with equality only if all d_j vanish. Finally, $w_j^\star - w_{j'}^\star = B^\star(d_{j'} - d_j)/\sigma^2$, which yields the monotonicity claims. □

Proof of Proposition 1. Under Assumption 1, $d_j = d_\infty\big(1 - e^{-m_0 \Delta_j}\big)$, and substituting into Equation (6) gives

$$w_j^\star = \alpha + \beta \, e^{-m_0 \Delta_j}, \qquad \alpha = \frac{\mu - B^\star d_\infty}{\sigma^2}, \qquad \beta = \frac{B^\star d_\infty}{\sigma^2}.$$

The ages are distinct, so $\Delta_{n-1} > 0$ and $d_{n-1} > 0$; Lemma 1 then gives $B^\star > 0$ and hence $\beta > 0$. Thus $w_j^\star$ decreases strictly with age, and its age-dependent part $w_j^\star - \alpha = \beta \, e^{-m_0 \Delta_j}$ decays at the rate m_0. □

Sign of the offset. Substituting $B^\star$ from Equation (6) into α gives

$$\alpha = \frac{\mu\big[\sigma^2 - \sum_k d_k (d_\infty - d_k)\big]}{\sigma^2\big(\sigma^2 + \sum_k d_k^2\big)}.$$

Because $\mu > 0$ and $0 \leq d_k \leq d_\infty$, the offset has the sign of $\sigma^2 - \sum_k d_k(d_\infty - d_k)$. It is positive when noise dominates, so that the optimum keeps a floor weight on every chunk, and negative when drift dominates. The pure exponential $w_k \propto e^{-m_0 \Delta_k}$ is exact only when the two terms balance, which is why RTE treats it as a one-parameter approximation with a free rate m rather than as the minimizer.

Sign constraints. When $\alpha < -\beta e^{-m_0 \Delta_{n-1}}$, the stalest chunk receives negative weight and the aggregate extrapolates away from it. If the constraints $w_j \geq 0$ are imposed as well, the Karush–Kuhn–Tucker conditions give $w_j = \max\{0, (\tilde{\mu} - \tilde{B} d_j)/\sigma^2\}$, where $\tilde{B} = \boldsymbol{d}^\top \boldsymbol{w} \geq 0$ and $\tilde{\mu}$ is fixed by $\mathbf{1}^\top \boldsymbol{w} = 1$. Because d is non-decreasing, the zero weights form a tail of the oldest chunks. RTE keeps the exponential component of Proposition 1 and drops the offset, which keeps every weight positive.

A.2 The Exponential Family

RTE restricts the weights to $w_k(m) = e^{-m\Delta_k} / \sum_l e^{-m\Delta_l}$ with $m \geq 0$. We write $\mathrm{MSE}(m) := \mathrm{MSE}(\boldsymbol{w}(m))$ and denote its derivative with respect to m by MSE'. The weights are uniform at $m = 0$ and concentrate on the freshest chunk as $m \to \infty$, so $\mathrm{MSE}(m) \to \mathrm{MSE}(\infty) := d_0^2 + \sigma^2$. Averages over the active chunks are written $\bar{\Delta} = \frac{1}{n}\sum_k \Delta_k$ and $\bar{d} = \frac{1}{n}\sum_k d_k$, and $\mathrm{Cov}(\Delta, d) := \frac{1}{n}\sum_k (\Delta_k - \bar{\Delta})(d_k - \bar{d})$ is their population covariance.

Proposition 2 (Neither limit is optimal)**.** *Let $n \geq 2$ and $\sigma^2 > 0$, and suppose that d is non-decreasing and not constant on $\{\Delta_0, \ldots, \Delta_{n-1}\}$.*

(i) $\mathrm{MSE}'(0) = -2\,\bar{d}\,\mathrm{Cov}(\Delta, d) < 0$.

(ii) With $\Delta_k = \tau + ks$ and $q := e^{-ms}$, $\mathrm{MSE}(m) = \mathrm{MSE}(\infty) - 2\big[\sigma^2 - d_0(d_1 - d_0)\big]q + O(q^2)$ *as $m \to \infty$.*

(iii) If $\sigma^2 > d_0(d_1 - d_0)$, which always holds at a replanning instant, where $\tau = 0$ and $d_0 = 0$, then MSE *attains its infimum over $[0, \infty)$ at a finite $m^\star > 0$, and* $\mathrm{MSE}(m^\star) < \min\{\mathrm{MSE}(0), \mathrm{MSE}(\infty)\}$.

Proof. (i) Differentiating $w_k(m)$ at $m = 0$, where every weight equals $1/n$, gives

$$w_k'(0) = \frac{-n\Delta_k + \sum_l \Delta_l}{n^2} = \frac{\bar{\Delta} - \Delta_k}{n}.$$

These derivatives sum to zero, so the variance term is stationary at the origin: its derivative is $2\sigma^2 \sum_k w_k(0) w_k'(0) = \frac{2\sigma^2}{n}\sum_k w_k'(0) = 0$. For the bias, $B(0) = \bar{d}$ and $B'(0) = \sum_k w_k'(0) d_k = \frac{1}{n}\sum_k (\bar{\Delta} - \Delta_k) d_k = -\mathrm{Cov}(\Delta, d)$, so $\mathrm{MSE}'(0) = 2B(0)B'(0) = -2\bar{d}\,\mathrm{Cov}(\Delta, d)$. For the sign, note that

$$2n^2\,\mathrm{Cov}(\Delta, d) = \sum_{j,k} (\Delta_j - \Delta_k)(d_j - d_k).$$

Every summand is non-negative, because the ages increase with the index and d is non-decreasing, and a pair with $d_j \neq d_k$ contributes a positive summand, which exists because d is not constant on the active ages. Hence $\mathrm{Cov}(\Delta, d) > 0$, and $\bar{d} > 0$ because $d \geq 0$ is not identically zero there.

(ii) With $S = \sum_{l<n} q^l = 1 + q + O(q^2)$, the weights are $w_0 = 1 - q + O(q^2)$, $w_1 = q + O(q^2)$, and $w_k = O(q^2)$ for $k \geq 2$. Therefore $B = d_0 + (d_1 - d_0)q + O(q^2)$, so that $B^2 = d_0^2 + 2d_0(d_1 - d_0)q + O(q^2)$, and $\|\boldsymbol{w}\|^2 = 1 - 2q + O(q^2)$. Adding the two terms of Equation (5) gives the expansion.

(iii) $\mathrm{MSE}(m)$ is continuous on $[0, \infty)$ and tends to $\mathrm{MSE}(\infty)$. By (ii) and the assumed inequality, $\mathrm{MSE}(m_1) < \mathrm{MSE}(\infty)$ for some finite m_1, and hence there is an $M > m_1$ with $\mathrm{MSE}(m) > \mathrm{MSE}(m_1)$ for all $m \geq M$. The minimum of MSE over the compact interval $[0, M]$ is attained at some $m^\star$ and does not exceed $\mathrm{MSE}(m_1)$, so it is also the infimum over $[0, \infty)$ and lies strictly below $\mathrm{MSE}(\infty)$. By (i), $\mathrm{MSE}(m) < \mathrm{MSE}(0)$ for all sufficiently small $m > 0$, so $m^\star > 0$ and $\mathrm{MSE}(m^\star) < \mathrm{MSE}(0)$. □

At an intermediate step the freshest chunk has already drifted by $d_0 = d(\tau)$, and blending in the next chunk pays off to first order in q only if the noise variance exceeds $d(\tau)\big[d(\tau + s) - d(\tau)\big]$. At a replanning instant the right-hand side vanishes, so the interior optimum requires nothing beyond $\sigma^2 > 0$.

A.3 Nested Plan Innovations

Assumption 1 treats the stale-plan error as a deterministic function of age. The error can instead accumulate: the plan that a chunk commits to at its query misses every innovation that arrives afterwards, so an older chunk misses all the innovations that a younger one misses and more. With innovations of variance ν per control step and independent prediction noise of variance σ^2, the errors e_k of chunks with ages $\Delta_k = \tau + ks$, $k = 0, \ldots, n-1$, have covariance $\mathrm{Cov}(e_j, e_k) = \nu \min(\Delta_j, \Delta_k) + \sigma^2 \delta_{jk}$.

Proposition 3 (Near-exponential optimum)**.** *Under this model the weights minimizing the mean squared error subject to* $\sum_k w_k = 1$ *are*

$$w_k^\star \propto r^k + r^{2n-1-k}, \qquad r = 1 + \tfrac{\lambda}{2} - \sqrt{\lambda + \tfrac{\lambda^2}{4}} \in (0,1), \qquad \lambda = \frac{\nu s}{\sigma^2}.$$

They are positive, do not depend on τ*, and, apart from the end-point term* r^{2n-1-k}*, which matters mainly for the oldest chunks, decay exponentially with age at the rate* $m^\star = -\ln r/s$*, with* $m^\star \approx \sqrt{\nu/(\sigma^2 s)}$ *for small* λ*.*

Proof. The term $\nu\tau$ is common to all entries of the covariance and does not change a minimizer constrained to $\sum_k w_k = 1$, so we may take $\tau = 0$. The Lagrange condition is $\nu s \sum_j \min(j,k)\, w_j + \sigma^2 w_k = \mu$ for every k. Its first difference in k gives $\nu s \sum_{j>k} w_j + \sigma^2(w_{k+1} - w_k) = 0$ for $0 \le k \le n-2$, which at $k = n-2$ reads $w_{n-2} = (1+\lambda)\, w_{n-1}$, and differencing once more gives $w_{k+1} - (2+\lambda)w_k + w_{k-1} = 0$ for $0 < k < n-1$. The roots of $x^2 - (2+\lambda)x + 1$ are r and $1/r$, and since $r + 1/r = 2 + \lambda$, the end condition holds exactly for $w_k \propto r^k + r^{2n-1-k}$, which is positive. Normalizing gives the stated weights. □

This is the optimality of exponential smoothing for a random walk observed in noise (Muth, 1960), read backward in time: each chunk reports the plan at its own query time. The optimal rate grows with the innovation-to-noise ratio and falls with the stride, as $s^{-1/2}$ for small λ, whereas the success-optimal rate of π_0 in Figure 6 does not fall with the stride and stays between 0.05 and 0.2 from s=2 to s=40. A noise component shared by chunks conditioned on nearby observations, which the repeated-draw analysis of Appendix A.4 indicates, would tie the useful window to physical time rather than to the number of chunks.

Two overlapping chunks. When $s = H/2$, every step blends exactly two chunks, of ages τ and $\tau + s$, with weights $w_0 = 1/(1 + e^{-ms})$ and $w_1 = 1 - w_0$. Under Equation (4) with a linear drift $d(\Delta) = g\Delta$, blending lowers the error averaged over the stride if and only if $g^2\, \mathbb{E}_\tau\big[(\tau + w_1 s)^2 - \tau^2\big] < 2 w_0 w_1 \sigma^2$, which at m=0.1 and s=5 reads $g/\sigma < 0.21$: with only two chunks, a drift that grows by a fifth of the noise level per step already cancels the benefit of blending.

A.4 Correlated and Age-Dependent Noise

Equicorrelated noise. Suppose that $\mathrm{Cov}(\boldsymbol{\varepsilon}) = \sigma^2\big[(1-\rho)I + \rho\, \mathbf{1}\mathbf{1}^\top\big]$, which is positive semidefinite exactly when $-1/(n-1) \le \rho \le 1$. For $\mathbf{1}^\top \boldsymbol{w} = 1$,

$$\mathrm{Var}\Big(\sum_k w_k \varepsilon_k\Big) = \sigma^2\big[(1-\rho)\|\boldsymbol{w}\|^2 + \rho\,(\mathbf{1}^\top \boldsymbol{w})^2\big] = (1-\rho)\sigma^2\|\boldsymbol{w}\|^2 + \rho\sigma^2.$$

The MSE is therefore Equation (5) with σ^2 replaced by $(1-\rho)\sigma^2$, plus the constant $\rho\sigma^2$, which does not depend on the weights. For $\rho < 1$, Lemma 1, Proposition 1, and part (i) of Proposition 2 hold unchanged, and part (iii) holds when $(1-\rho)\sigma^2 > d_0(d_1 - d_0)$. A shared noise component thus lowers the error that aggregation can remove without changing the form of the optimal weights. For $\rho = 1$ averaging removes no noise, and no weighting improves on the freshest chunk at a replanning instant.

Age-dependent noise. If instead the ε_k are uncorrelated with variances $\sigma_k^2 > 0$, the Lagrange condition becomes $B d_j + \sigma_j^2 w_j = \mu$, so that $w_j^\star = (\mu - B^\star d_j)/\sigma_j^2$, and the argument of Lemma 1, with the weighted Cauchy–Schwarz inequality, again gives $\mu > 0$ and $B^\star \ge 0$. When σ_k^2 is non-decreasing in age, older chunks are down-weighted both for their drift and for their larger variance, although the minimizer is then no longer affine in $e^{-m_0 \Delta_j}$. Along the exponential family the variance term contributes $\frac{2}{n}\sum_k w_k'(0)\sigma_k^2 = -\frac{2}{n}\mathrm{Cov}(\Delta, \sigma^2) \le 0$ to the derivative at the origin, so part (i) of Proposition 2 becomes $\mathrm{MSE}'(0) = -2\bar{d}\,\mathrm{Cov}(\Delta, d) - \frac{2}{n}\mathrm{Cov}(\Delta, \sigma^2) < 0$.

Repeated draws on one observation. The noise ε_k of Equation (4) combines the sampling noise of a draw with the error that the policy makes on the observation that the chunk was conditioned on. Several draws on the same observation share the second component, so their noise is equicorrelated

Algorithm 1 RTE at one control step

Require: current observation o_t, instruction ℓ, buffer $\mathcal{B}$, step t, stride s, decay m, horizon H
Ensure: action $\hat{a}_t$ to execute, updated buffer $\mathcal{B}$

if $t \bmod s = 0$ **then** ▷ replan at fixed stride
  $z \sim p(z)$; $\boldsymbol{a}^{(t)} \leftarrow G_\theta(o_t, \ell, z)$
  append $(t, \boldsymbol{a}^{(t)})$ to $\mathcal{B}$ ▷ cache chunk with its generation time
end if
remove every $(t_k, \boldsymbol{a}^{(t_k)})$ with $t_k + H \le t$ from $\mathcal{B}$ ▷ discard expired chunks
$\mathcal{K}_t \leftarrow \{k : t_k \le t < t_k + H\}$ ▷ retrieve active chunks covering step t
for $k \in \mathcal{K}_t$ **do**
  $\Delta_{k,t} \leftarrow t - t_k$; $w_k \leftarrow e^{-m\Delta_{k,t}}$ ▷ compute age-decayed weights
end for
$\hat{a}_t \leftarrow \big(\sum_{k\in\mathcal{K}_t} w_k a_t^{(t_k)}\big) / \sum_{k\in\mathcal{K}_t} w_k$ ▷ temporal aggregation
return $\hat{a}_t$

with ρ equal to its share of σ^2, and by the result above, averaging them removes only the sampling part. Chunks conditioned on successive observations share this component only in part. This is consistent with Appendix C.3, where averaging four draws per query recovers little of the gain that blending successive chunks achieves, and blending still helps when all chunks share one source-noise draw.

A.5 Measuring Drift and Sampling Noise

Figure 1b is recorded during $1{,}500$ rollouts of RTE on π_0-Long at $s{=}3$ and $m{=}0.1$, 500 for each of seeds 7, 8, and 9. At every control step, each active chunk contributes the RMS over the six pose coordinates of its gap to the freshest chunk, and the figure averages these values by age; chunks younger than the stride are the freshest chunk itself and are omitted. At every replanning instant the policy is queried a second time on the same observation, and the RMS pose gap between the two draws is recorded at each chunk index; this second chunk is never executed. At age 3, the gap to the freshest chunk is 0.059, against 0.042 between two draws on the same observation, and by age 48 the former reaches 0.131 while the latter rises only to 0.051. On $\pi_{0.5}$-Long, whose chunks span ten steps, the same protocol gives a gap to the freshest chunk that grows from 0.035 at age 3 to 0.052 at age 9, while two draws on the same observation differ by 0.023 to 0.025 throughout.

Under Equation (4), the gap to the freshest chunk contains the sampling noise of both predictions in addition to the drift difference, so it stays well above zero even at the shortest age gap. The squared gaps are heavy-tailed: the largest one percent of rows carries about two thirds of their sum, and most of these rows come from failed episodes.

B Experimental Details

B.1 Implementation

RTE runs on the client, between the policy server and the robot controller, and never touches the policy weights. Algorithm 1 gives one control step: every s steps it queries the policy and caches the new chunk with its generation time, drops chunks whose horizon has passed, and blends the predictions that the remaining chunks make for the current step. On SIMPLER, the buffer keeps only the two most recent chunks, as CogACT's released evaluation does.

B.2 Model Architectures and Checkpoints

We evaluate publicly released checkpoints wherever they exist, namely π_0, $\pi_{0.5}$, and GR00T-N1.7 on LIBERO and CogACT on SIMPLER and LIBERO, and otherwise fine-tune from the public base models following their official protocols, namely π_0, $\pi_{0.5}$, and GR00T-N1.7 on RoboTwin and all real-world Franka policies. Table 5 lists the artifacts and Table 6 the recipes; custom models use the final scheduled checkpoint. Within every comparison, all deployments load the same checkpoint, so RTE and its controls differ only in how the policy is deployed and change no learned weights.

Table 5: **Evaluated checkpoints and public artifact locations.** Official releases are evaluated without further training on their respective benchmarks. "Custom Fine-tuned" denotes policies trained by us from the listed public base models, following recipes in Table 6.

Model	Evaluation Suite	Checkpoint Origin	Public Artifact Path / Revision
π_0	LIBERO	Official Release	`gs://openpi-assets/checkpoints/pi0_libero`
$\pi_{0.5}$	LIBERO	Official Release	`gs://openpi-assets/checkpoints/pi05_libero`
π_0/$\pi_{0.5}$	RoboTwin	Custom Fine-tuned	Base: `gs://openpi-assets/checkpoints/pi0{,5}_base`
GR00T-N1.7	LIBERO	Official Release	HF `nvidia/GR00T-N1.7-LIBERO`
GR00T-N1.7	RoboTwin	Custom Fine-tuned	Base: HF `nvidia/GR00T-N1.7-3B`
CogACT-Base	SIMPLER	Official Release	HF `CogACT/CogACT-Base` (rev. `6550bf09`)
DB-CogACT	LIBERO	Official Release	HF `Dexmal/libero-db-cogact`
π_0/$\pi_{0.5}$	Franka	Custom Fine-tuned	Base: `gs://openpi-assets/checkpoints/pi0{,5}_base`
GR00T-N1.7	Franka	Custom Fine-tuned	Base: HF `nvidia/GR00T-N1.7-3B`
CogACT	Franka	Custom Fine-tuned	Base: HF `CogACT/CogACT-Base` (rev. `6550bf09`)

Table 6: **Training configurations for custom fine-tuned policies**, separated by Simulation and Real-World (Franka) deployments. $H/O/A$ denote prediction, observation, and training action horizons; deployment strides are reported separately in Table 7. Only the RoboTwin π_0/$\pi_{0.5}$ models utilize EMA weights during evaluation.

Hyperparameter	Simulation Fine-tuning		Real-World (Franka) Fine-tuning		
	π_0/$\pi_{0.5}$ (RoboTwin)	GR00T-N1.7 (RoboTwin)	π_0/$\pi_{0.5}$	GR00T-N1.7	CogACT
Training Scope	One policy for 50 tasks	One policy for 50 tasks	One policy across 5 tasks	One policy across 5 tasks	One policy across 5 tasks
Fine-tuning Type	LoRA	Full, VLM frozen	Full	Full, VLM frozen	Full
$H/O/A$	50/1/50	16/1/16	50/1/50	40/1/40	16/1/16
Camera Inputs	Head + two wrist RGB	Head + two wrist RGB	Top RGB	Top RGB	Top RGB
Action Space	Joint positions, arms relative to state in training		EEF pose relative to the chunk's first state + absolute gripper (standardized)		
Visual Encoder	Released VLM	Released VLM	Released VLM	Released VLM	Released DINOv2+SigLIP
Image Size	224×224	224×224	224×224	256×256	224×224
Data Augmentation	Crop, rotation, color jitter	Color jitter	Crop, rotation, color jitter	Crop, color jitter	Crop, color jitter
Hardware & Precision	8× A100 (bf16)	8× A100 (bf16)	8× A100 (bf16)	8× A100 (bf16)	8× A100 (bf16)
Batch Size	32	128 (16/GPU)	32	64 (8/GPU)	256 (32/GPU)
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Peak LR	2.5×10^{-5}	10^{-4}	2.5×10^{-5}	10^{-4}	2×10^{-5}
LR Schedule	Cosine to 2.5×10^{-6}	Cosine	Cosine to 2.5×10^{-6}	Cosine	Constant
Warmup	1000 steps	5% steps	1000 steps	5% steps	None
Weight Decay	10^{-10}	10^{-5}	10^{-10}	10^{-5}	0
Gradient Clipping	1.0	1.0	1.0	1.0	1.0
EMA	0.99	Disabled	Disabled	Disabled	Disabled
Training Seed	0	0	42	42	42
Final Checkpoint	30k steps	20k steps	30k steps	30k steps	30 epochs (12,000 steps)

B.3 Per-Backbone Deployment Details

Table 7 details the deployment configurations for each evaluated backbone and benchmark. For each policy, we list the generative framework, the prediction horizon H, and the released deployment, which is what Base denotes throughout: fixed-prefix execution at the released stride on LIBERO, whole-chunk open-loop execution on RoboTwin and the Franka arm, and CogACT's AAE on SIMPLER. RTE and its controls replan at matched strides, either the released one or shorter ones that create overlap. Actions are end-effector pose increments on LIBERO and SIMPLER and joint positions on RoboTwin, each with a continuous gripper command; for the small rotation increments of one control step, averaging each coordinate agrees with the rotational mean to first order.

ACT's Temporal Ensembling. As noted in Section 2, ACT aligns overlapping chunks by absolute temporal target and, in both its paper and its released implementation, applies an exponential weighting scheme that prioritizes older queries. Specifically, it attributes the highest weighting coefficient to the *oldest* valid action prediction, with the decay rate controlling how quickly new observations are incorporated. In the parametrization of Equation (3), this rule corresponds to $m = -0.01/s$, the opposite direction to RTE, which favors the freshest prediction.

B.4 The Cost of Aggregation

While Section 4 establishes the theoretical $O(KD)$ complexity of the temporal blend, this section provides an empirical analysis of its wall-clock deployment cost. The evaluation harness records every policy invocation and its corresponding latency, allowing us to precisely measure both the query frequency and the execution time per query. Table 8 summarizes these metrics for the six

Table 7: **Deployment configuration of each backbone and benchmark.** H is the prediction horizon, and the released deployment is the rule of the public release or of the official fine-tuning recipe, which Base denotes throughout. DB-CogACT commits $12/16/16/15$ actions by suite in its release, which reaches $93.5/98.1/95.3/91.5\%$ on Spatial/Object/Goal/Long, 94.6% on average over seeds $7/8/9$; we evaluate both fixed-prefix execution and RTE at $s = 8$ and $s = 4$. RTE uses $m = 0.1$ throughout.

Model	Generative Framework	Evaluation Suite	Checkpoint Scope	Horizon	Released Deployment (Base)	RTE Stride
DB-CogACT	Diffusion (DDIM)	LIBERO	Official Release	$H = 16$	Fixed-prefix execution, 12–16 actions	$s \in \{8, 4\}$
CogACT-Base	Diffusion (DDIM)	SIMPLER	Official Release	$H = 16$	Released AAE, $s = 1$, horizon 2	$s = 1$
π_0	Flow Matching	LIBERO	Official Release	$H = 50$	Fixed-prefix execution, $s = 5$	$s \in \{5, 3\}$
$\pi_{0.5}$	Flow Matching	LIBERO	Official Release	$H = 10$	Fixed-prefix execution, $s = 5$	$s \in \{5, 3\}$
$\pi_0/\pi_{0.5}$	Flow Matching	RoboTwin	Custom (50-task)	$H = 50$	Open-loop chunk, $s = H$	$s \in \{10, 25\}$
GR00T-N1.7	Flow Matching	LIBERO	Official Release	$H = 16$	Execute 8 actions, $s = 8$	$s \in \{8, 4\}$
GR00T-N1.7	Flow Matching	RoboTwin	Custom (50-task)	$H = 16$	Open-loop chunk, $s = H$	$s \in \{4, 8\}$
CogACT	Diffusion (DDIM)	Franka	Custom (5-task)	$H = 16$	Open-loop chunk, $s = H$	$s = 8$
$\pi_0/\pi_{0.5}$	Flow Matching	Franka	Custom (5-task)	$H = 50$	Open-loop chunk, $s = H$	$s = 10$
GR00T-N1.7	Flow Matching	Franka	Custom (5-task)	$H = 40$	Open-loop chunk, $s = H$	$s = 10$

Table 8: **Empirical deployment cost on π_0-Long** over seeds $7/8/9$ (500 episodes per seed, eight A100 GPUs). Success, calls per episode, and query ratio are mean $\pm$ population std across seeds; pooled calls are totals over all $1{,}500$ episodes; latency is the pooled mean $\pm$ std across calls of the policy call alone. The open-loop latency is slightly higher because its $9{,}734$ calls include JIT warm-up.

Configuration	Success (%)	Total Calls	Calls / Ep.	Query Ratio	Per-call Latency (ms)
Open-loop (s=50)	66.4 ± 1.9	9 734	6.5 ± 0.1	$0.107 \pm 0.001\times$	84.94 ± 5.42
Base (s=5)	80.7 ± 0.5	91 125	60.8 ± 0.4	$1.000 \pm 0.000\times$	73.50 ± 3.85
Freshest-only (s=1)	73.1 ± 1.9	505 281	336.9 ± 6.8	$5.544 \pm 0.075\times$	72.27 ± 4.26
RTE (s=1)	88.5 ± 1.0	419 500	279.7 ± 2.3	$4.604 \pm 0.036\times$	72.34 ± 3.52
Equal weights (s=3)	81.9 ± 1.9	148 802	99.2 ± 1.8	$1.633 \pm 0.026\times$	72.51 ± 1.53
RTE (s=3)	89.9 ± 0.9	138 432	92.3 ± 1.4	$1.519 \pm 0.026\times$	72.55 ± 2.20

estimator configurations evaluated on π_0-Long, all sharing identical hardware and the same $1{,}500$ episode schedule across three seeds.

Computational Overhead of the Temporal Blend. The blend runs on the client after each policy call returns. Timed on its own, the weighted mean takes about 21 μs per control step for the 17 chunks active at s=3 and 39 μs for the 50 chunks active at s=1, against roughly 72 ms per policy call. The per-call latencies in Table 8 stay between 72.3 and 72.6 ms for every aggregating configuration and at 72.27 ms for the Freshest-only baseline, so the policy call itself is unaffected and the cost of RTE is set by its query rate.

Query Rate and Replanning Stride. The replanning stride determines the nominal query cadence, while realized call counts also depend on how quickly an episode terminates. Shortening the stride from the released s=5 to s=3 produces a measured $1.52\times$ call ratio, while s=1 produces $4.60\times$ to $5.54\times$. A matched-stride comparison therefore fixes the query schedule and per-query computation, although successful configurations can issue fewer calls by ending earlier. Equal weighting and RTE at s=3, for example, share the same cadence; their realized call totals differ by 7%, while their success rates are 81.9% and 89.9%, respectively.

C Additional Experimental Results

This appendix collects every supplementary experiment referenced from the main text: finite-N headroom diagnostics and episode-level recovery attribution; full per-task RoboTwin results with structural controls; and a three-seed replication of the estimator ablation together with alternative weighting and sampling mechanisms.

Table 9: **Finite-N LIBERO oracle references and descriptive recovery ratio** (mean $\pm$ population std over seeds $7/8/9$). pass@1 is Base at the released stride, and RTE runs at s=3. The ratio is the fraction of observed pass@4 headroom recovered online: $(\text{RTE} - \text{pass@1})/(\text{pass@4} - \text{pass@1})$.

Model	Suite	pass@1	RTE	pass@4	pass@8	Ratio
π_0	Long	80.7 ± 0.5	89.9 ± 0.9	94.6 ± 0.2	96.1 ± 0.3	0.67
	Goal	93.7 ± 0.9	97.0 ± 0.3	99.9 ± 0.1	100.0 ± 0.0	0.53
	Spatial	96.9 ± 0.4	97.9 ± 0.2	100.0 ± 0.0	100.0 ± 0.0	0.33
	Object	97.3 ± 0.4	98.7 ± 0.5	100.0 ± 0.0	100.0 ± 0.0	0.51
$\pi_{0.5}$	Long	92.5 ± 0.7	94.5 ± 0.7	98.7 ± 0.3	99.3 ± 0.1	0.31
	Goal	97.9 ± 0.6	97.9 ± 0.4	99.7 ± 0.1	99.9 ± 0.1	0.04
	Spatial	98.2 ± 0.3	98.3 ± 0.8	100.0 ± 0.0	100.0 ± 0.0	0.04
	Object	98.5 ± 0.4	99.5 ± 0.2	100.0 ± 0.0	100.0 ± 0.0	0.64

Table 10: **Episode-level attribution on π_0-Long** over seeds $7/8/9$. Counts pool $1{,}500$ paired episodes; RTE runs at s=3, and the four pass@4 attempts are drawn independently of the Base rollout. The two right columns replace RTE with a further, independent rollout of Base on the same episodes. The gain of RTE over Base, 9.3 points, has a 95% paired bootstrap interval of $[6.9, 11.7]$ when the three seeds of each initial state are resampled together.

	RTE	Base rollout 2	Base rollout 3
Base fails		290	
of which pass@4 solves		212	
Newly solves	202	139	130
of which pass@4 also solves	169 (79.7%)	130 (61.3%)	126 (59.4%)
of which pass@4 never solves	33 (42.3%)	9 (11.5%)	4 (5.1%)
Regresses	63	118	134

C.1 Diagnosing Stochastic Headroom

We first report the finite-N oracle across all eight LIBERO model–suite pairs, then use paired episodes on π_0-Long to identify which failures RTE recovers. The paired analysis then tests whether RTE preferentially recovers resampling-solvable failures.

Finite-N oracle across suites. Table 9 shows that the available headroom is highly uneven. π_0-Long opens 13.9 points between pass@1 and pass@4, whereas $\pi_{0.5}$-Spatial and $\pi_{0.5}$-Goal open only 1.8 points. With such small denominators, the two 0.04 entries mean only that little headroom remains. At pass@8, five cells reach 100.0 and $\pi_{0.5}$-Goal reaches 99.9, so these suites are effectively saturated, whereas the two Long suites retain the clearest residual headroom and also show the largest RTE gain of each model, 9.3 and 1.9 points.

Episode-level recovery attribution. Table 10 reports the paired episode analysis discussed in the main text; percentages are taken over the 212 failures that pass@4 solves and the 78 that it does not. A further Base rollout recovers mostly the former, as resampling should, but far fewer of either kind than RTE, and it loses about twice as many Base successes. The remaining 33 newly solved cases are not solved by any of the four resampled Base rollouts. A blended trajectory need not coincide with any member of that finite sample set, so pass@4 does not enumerate every trajectory available to an online temporal estimator. Finite-N pass@N is therefore a reference point that an online estimator can exceed.

C.2 Simulation Results: RoboTwin 2.0

Tables 11 to 13 and Figure 5 report the same task-level evidence in complementary forms. The tables give every success rate under the `demo_clean` protocol, while the figure converts one stride per backbone into sorted per-task gains. Together they show that the averages are not driven by a few outliers: π_0 and $\pi_{0.5}$ each improve 43 of 50 tasks, and the gains, although larger where Base is

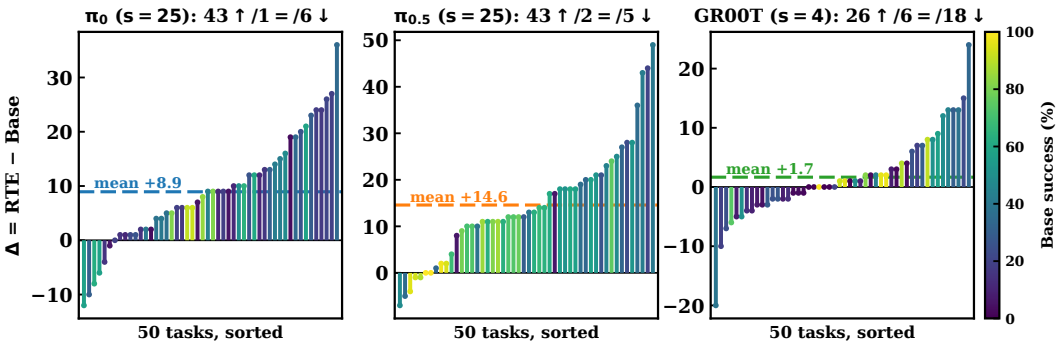


Figure 5: **Task-level view of Tables 11 to 13.** RTE−Base gains over the 50 tasks at one stride per backbone, $s = 25$ for $\pi_0/\pi_{0.5}$ and $s = 4$ for GR00T, sorted by gain and colored by Base success; panel titles count the tasks that improve (↑), stay level (=), or drop (↓), and dashed lines mark the mean.

weak, extend to tasks that Base already solves well. GR00T's improvements are more modest and concentrated around zero, consistent with Table 3, where faster replanning alone lowers its success.

Blend Contribution and ESS. Figure 4c plots the decomposition of Table 3 against the ESS of Section 4.3: the isolated blend contribution stays near zero at an ESS of 1.2 and becomes positive from 1.7 upward across π_0, $\pi_{0.5}$, and GR00T. The ESS is fixed by the horizon, the stride, and the decay rate, so the trend also reflects the freshest-only control, whose success falls at the shorter stride for π_0 and GR00T, where blending recovers the loss.

Where blending helps and where it hurts. Comparing RTE with the freshest-only control of Table 3 task by task gives the same picture as the averages. Blending raises success by more than ten points in 54 of the 300 task–stride pairs, most often at the settings with the largest ESS, namely 18 tasks for GR00T at s=4 and 16 for π_0 at s=10. It lowers success by as much in five pairs: $\pi_{0.5}$ on Move Stapler Pad, where success drops from 64% to 28%, and on Click Bell at s=10, π_0 on Place Empty Cup and Press Stapler at s=10, and π_0 on Put Bottles Dustbin at s=25. All five end in a precise placement or press. The steepest regressions against Base have a different source: π_0 on Click Alarmclock and Rotate QRCode and GR00T on Beat Block Hammer already lose more than ten points under the freshest-only control, which suggests that these tasks favor committing to a full chunk over frequent replanning rather than suffering from averaging.

Checkpoint maturity. Table 14 shows that doubling the training length raises Base from 33.9 to 38.2, and the gain of RTE at s=25 from +8.9 to +10.7.

C.3 Ablation Studies

Per-Seed Breakdown of the Estimator Ablation.

Table 4 reports the estimator means, and Table 15 gives their per-seed breakdown. Increasing the query rate without temporal aggregation lowers success below Base in every seed, and at each of the strides 1, 3, and 5, both limits of the weighting rule fall below the recency-weighted interior in every seed. Seed 7 is the only seed on which m was selected, and on seeds 8 and 9 RTE at s=3 exceeds Base by 10.6 and 8.6 points. The decomposition carries over to $\pi_{0.5}$-Long, whose freshest-only control at s=3 reaches 91.3%, below its Base of 92.5%, whereas RTE attains 94.5% at the same stride.

Joint Sweep of Decay Rate and Stride. Figure 6 sweeps the decay rate jointly with the stride on all four backbones, up to the chunk length. On π_0, uniform weights lose success steadily as the stride grows and the blended predictions age, whereas the best rate stays between 0.05 and 0.2 up to s=40 and improves on m=0.1 by less than two points; only near s=H, where the chunks barely overlap, does no rate improve on the freshest-only control. Where predictions stay young, as with the ten-step chunks of $\pi_{0.5}$, or where the policy is sensitive to single-draw variance, as with CogACT, uniform weights come close to the best rate or reach it.

Table 11: **Per-task RoboTwin 2.0 results of π_0** under the `demo_clean` protocol, success rates (%) over 100 rollouts per task. Base executes each chunk open-loop; Freshest, Equal, and RTE share the query schedule at each stride.

Task	Base	s=25			s=10		
		Freshest	Equal	RTE	Freshest	Equal	RTE
Adjust Bottle	40.0	42.0	42.0	42.0	42.0	44.0	44.0
Beat Block Hammer	64.0	72.0	55.0	74.0	70.0	74.0	85.0
Blocks Ranking RGB	18.0	34.0	35.0	42.0	12.0	35.0	43.0
Blocks Ranking Size	5.0	7.0	0.0	14.0	15.0	7.0	13.0
Click Alarmclock	57.0	43.0	47.0	45.0	24.0	32.0	28.0
Click Bell	40.0	46.0	27.0	44.0	23.0	10.0	16.0
Dump Bin Bigbin	58.0	79.0	80.0	79.0	60.0	87.0	83.0
Grab Roller	84.0	88.0	89.0	92.0	63.0	95.0	92.0
Handover Block	11.0	15.0	21.0	20.0	2.0	20.0	14.0
Handover Mic	49.0	65.0	62.0	59.0	42.0	77.0	73.0
Hanging Mug	5.0	15.0	7.0	12.0	8.0	20.0	10.0
Lift Pot	26.0	50.0	42.0	46.0	58.0	60.0	67.0
Move Can Pot	21.0	45.0	37.0	47.0	34.0	45.0	45.0
Move Pillbottle Pad	19.0	55.0	45.0	46.0	49.0	27.0	48.0
Move Playingcard Away	42.0	61.0	49.0	56.0	57.0	52.0	54.0
Move Stapler Pad	6.0	1.0	2.0	5.0	2.0	0.0	1.0
Open Laptop	15.0	14.0	19.0	16.0	11.0	22.0	12.0
Open Microwave	37.0	33.0	41.0	42.0	10.0	43.0	29.0
Pick Diverse Bottles	38.0	44.0	40.0	42.0	50.0	48.0	43.0
Pick Dual Bottles	34.0	47.0	40.0	53.0	58.0	52.0	62.0
Place A2B Left	16.0	25.0	22.0	22.0	16.0	21.0	24.0
Place A2B Right	11.0	24.0	15.0	21.0	8.0	18.0	18.0
Place Bread Basket	26.0	38.0	30.0	38.0	40.0	40.0	41.0
Place Bread Skillet	19.0	29.0	26.0	25.0	29.0	27.0	29.0
Place Burger Fries	29.0	51.0	30.0	52.0	70.0	50.0	61.0
Place Can Basket	30.0	43.0	43.0	43.0	34.0	39.0	41.0
Place Cans Plasticbox	17.0	46.0	29.0	41.0	33.0	31.0	40.0
Place Container Plate	81.0	96.0	77.0	90.0	96.0	85.0	93.0
Place Dual Shoes	16.0	30.0	12.0	29.0	31.0	30.0	32.0
Place Empty Cup	57.0	55.0	52.0	51.0	64.0	54.0	46.0
Place Fan	11.0	21.0	19.0	23.0	11.0	21.0	22.0
Place Mouse Pad	3.0	7.0	7.0	5.0	5.0	4.0	6.0
Place Object Basket	43.0	45.0	54.0	52.0	43.0	59.0	55.0
Place Object Scale	15.0	23.0	17.0	17.0	20.2	25.0	34.0
Place Object Stand	48.0	63.0	62.0	64.0	67.0	75.0	71.0
Place Phone Stand	22.0	21.0	13.0	23.0	9.0	16.0	13.0
Place Shoe	42.0	62.0	61.0	57.0	41.0	60.0	61.0
Press Stapler	62.0	53.0	54.0	54.0	53.0	41.0	41.0
Put Bottles Dustbin	26.0	41.0	43.0	27.0	34.0	54.0	53.0
Put Object Cabinet	11.0	12.0	15.0	12.0	11.0	23.0	25.0
Rotate QRCode	30.0	14.0	24.0	20.0	1.0	20.0	2.0
Scan Object	15.0	16.0	16.0	11.0	25.0	20.0	17.0
Shake Bottle	92.0	96.0	93.0	98.0	93.0	97.0	96.0
Shake Bottle Horizontally	91.0	99.0	96.0	97.0	87.0	98.0	93.0
Stack Blocks Three	3.0	22.0	12.0	22.0	6.0	36.0	22.0
Stack Blocks Two	34.0	71.0	60.0	70.0	19.0	66.0	61.0
Stack Bowls Three	60.0	71.0	68.0	72.0	64.0	73.0	67.0
Stack Bowls Two	81.0	93.0	88.0	86.0	92.0	88.0	89.0
Stamp Seal	19.0	23.0	24.0	19.0	28.0	18.0	27.0
Turn Switch	18.0	25.0	28.0	27.0	29.0	34.0	28.0
Average	33.94	43.42	39.40	42.88	36.98	43.46	43.40

Decay Rate on $\pi_{0.5}$. The chunks of $\pi_{0.5}$ span only ten steps, so no blended prediction is older than nine steps, and the landscape in Figure 6b is flatter than that of π_0: uniform weights stay within a

Table 12: **Per-task RoboTwin 2.0 results of $\pi_{0.5}$** under the `demo_clean` protocol, success rates (%) over 100 rollouts per task. Base executes each chunk open-loop; Freshest, Equal, and RTE share the query schedule at each stride.

Task	Base	s=25			s=10		
		Freshest	Equal	RTE	Freshest	Equal	RTE
Adjust Bottle	86.0	99.0	94.0	97.0	99.0	100.0	100.0
Beat Block Hammer	81.0	89.0	92.0	92.0	89.0	96.0	95.0
Blocks Ranking RGB	42.0	87.0	81.0	85.0	92.0	79.0	94.0
Blocks Ranking Size	40.0	44.0	37.0	53.0	49.0	48.0	57.0
Click Alarmclock	72.0	86.0	81.0	84.0	70.0	85.0	74.0
Click Bell	48.0	41.0	31.0	41.0	80.0	58.0	52.0
Dump Bin Bigbin	88.0	92.0	96.0	87.0	96.0	98.0	93.0
Grab Roller	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Handover Block	33.0	45.0	51.0	53.0	22.0	79.0	72.0
Handover Mic	35.0	69.0	69.0	71.0	66.0	90.0	92.0
Hanging Mug	6.0	19.0	28.0	23.0	30.0	30.0	36.0
Lift Pot	73.0	90.0	91.0	85.0	92.0	88.0	88.0
Move Can Pot	33.0	59.0	53.0	60.0	62.0	69.0	66.0
Move Pillbottle Pad	40.0	86.0	78.0	89.0	96.0	86.0	93.0
Move Playingcard Away	73.0	88.0	87.0	83.0	96.0	82.0	91.0
Move Stapler Pad	6.0	13.0	13.0	14.0	64.0	8.0	28.0
Open Laptop	77.0	82.0	87.0	89.0	74.0	86.0	84.0
Open Microwave	57.0	77.0	81.0	77.0	58.0	79.0	83.0
Pick Diverse Bottles	63.0	78.0	75.0	76.0	87.0	74.0	82.0
Pick Dual Bottles	67.0	80.0	84.0	81.0	95.0	80.0	89.0
Place A2B Left	63.0	75.0	68.0	74.0	80.0	70.0	75.0
Place A2B Right	48.0	71.0	60.0	73.0	70.0	66.0	65.0
Place Bread Basket	61.0	87.0	72.0	78.0	86.0	80.0	91.0
Place Bread Skillet	57.0	79.0	70.0	75.0	88.0	79.0	80.0
Place Burger Fries	85.0	94.0	96.0	96.0	91.0	94.0	94.0
Place Can Basket	52.0	74.0	77.0	80.0	69.0	80.0	73.0
Place Cans Plasticbox	23.0	53.0	36.0	51.0	91.0	62.0	94.0
Place Container Plate	93.0	93.0	94.0	92.0	99.0	97.0	94.0
Place Dual Shoes	36.0	54.0	47.0	55.0	62.0	64.0	71.0
Place Empty Cup	95.0	93.0	88.0	91.0	98.0	95.0	96.0
Place Fan	44.0	59.0	60.0	67.0	42.0	59.0	75.0
Place Mouse Pad	29.0	40.0	34.0	41.0	43.0	43.0	47.0
Place Object Basket	60.0	73.0	66.0	78.0	73.0	76.0	83.0
Place Object Scale	64.0	88.0	75.0	78.0	85.0	80.0	91.0
Place Object Stand	79.0	89.0	82.0	88.0	91.0	87.0	93.0
Place Phone Stand	55.0	69.0	66.0	73.0	52.0	66.0	71.0
Place Shoe	63.0	66.0	67.0	74.0	64.0	70.0	81.0
Press Stapler	68.0	78.0	61.0	72.0	68.0	63.0	65.0
Put Bottles Dustbin	52.0	70.0	70.0	73.0	75.0	78.0	86.0
Put Object Cabinet	33.0	41.0	44.0	34.0	47.0	46.0	49.0
Rotate QRCode	51.0	59.0	61.0	69.0	21.0	61.0	60.0
Scan Object	26.0	47.0	38.0	47.0	43.0	50.0	48.0
Shake Bottle	100.0	99.0	99.0	100.0	99.0	100.0	100.0
Shake Bottle Horizontally	98.0	100.0	100.0	100.0	98.0	100.0	100.0
Stack Blocks Three	20.0	64.0	57.0	64.0	65.0	72.0	84.0
Stack Blocks Two	75.0	99.0	95.0	99.0	96.0	99.0	100.0
Stack Bowls Three	70.0	69.0	73.0	80.0	74.0	76.0	82.0
Stack Bowls Two	94.0	94.0	89.0	96.0	99.0	96.0	98.0
Stamp Seal	38.0	51.0	43.0	48.0	64.0	51.0	62.0
Turn Switch	35.0	33.0	31.0	30.0	42.0	38.0	56.0
Average	57.74	71.70	68.56	72.32	73.84	74.26	78.66

point of the best rate at every stride below H, and the rate m=0.1, fixed on π_0, within three points. The freshest-only control trails most at s=2, where it switches to a new noise draw every two steps.

Table 13: **Per-task RoboTwin 2.0 results of GR00T-N1.7** under the `demo_clean` protocol, success rates (%) over 100 rollouts per task. Base executes each chunk open-loop; Freshest, Equal, and RTE share the query schedule at each stride.

Task	Base	s=8			s=4		
		Freshest	Equal	RTE	Freshest	Equal	RTE
Adjust Bottle	91.0	91.0	94.0	98.0	79.0	100.0	99.0
Beat Block Hammer	39.0	9.0	46.0	25.0	0.0	20.0	19.0
Blocks Ranking RGB	0.0	0.0	0.0	1.0	0.0	0.0	0.0
Blocks Ranking Size	0.0	0.0	2.0	0.0	1.0	0.0	0.0
Click Alarmclock	96.0	93.0	99.0	97.0	88.0	99.0	97.0
Click Bell	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Dump Bin Bigbin	50.0	38.0	50.0	53.0	23.0	48.0	45.0
Grab Roller	92.0	94.0	91.0	92.0	86.0	97.0	93.0
Handover Block	2.0	4.0	4.0	3.0	0.0	4.0	1.0
Handover Mic	8.0	1.0	9.0	5.0	0.0	6.0	3.0
Hanging Mug	2.0	2.0	2.0	3.0	1.0	4.0	1.0
Lift Pot	8.0	4.0	19.0	21.0	0.0	11.0	11.0
Move Can Pot	9.0	18.0	15.0	18.0	20.0	19.0	10.0
Move Pillbottle Pad	14.0	5.0	24.0	16.0	2.0	17.0	10.0
Move Playingcard Away	38.0	38.0	49.0	50.0	27.0	44.0	51.0
Move Stapler Pad	3.0	1.0	2.0	2.0	0.0	3.0	5.0
Open Laptop	43.0	31.0	44.0	42.0	23.0	38.0	45.0
Open Microwave	0.0	1.0	0.0	0.0	8.0	1.0	0.0
Pick Diverse Bottles	51.0	48.0	52.0	50.0	45.0	51.0	52.0
Pick Dual Bottles	44.0	57.0	61.0	58.0	43.0	69.0	57.0
Place A2B Left	27.0	27.0	36.0	32.0	18.0	34.0	25.0
Place A2B Right	29.0	23.0	30.0	32.0	12.0	32.0	26.0
Place Bread Basket	49.0	50.0	53.0	51.0	37.0	54.0	61.0
Place Bread Skillet	25.0	22.0	36.0	33.0	18.0	25.0	31.0
Place Burger Fries	82.0	78.0	82.0	88.0	70.0	86.0	84.0
Place Can Basket	13.0	14.0	13.0	23.0	20.0	12.0	11.0
Place Cans Plasticbox	30.0	19.0	42.0	30.0	21.0	31.0	43.0
Place Container Plate	89.0	91.0	93.0	91.0	73.0	96.0	93.0
Place Dual Shoes	3.0	2.0	1.0	3.0	1.0	4.0	2.0
Place Empty Cup	57.0	63.0	64.0	64.0	64.0	66.0	66.0
Place Fan	0.0	2.0	3.0	5.0	1.0	1.0	1.0
Place Mouse Pad	7.0	2.0	8.0	5.0	1.0	7.0	4.0
Place Object Basket	24.0	20.0	21.0	19.0	24.0	21.0	17.0
Place Object Scale	19.0	17.0	23.0	17.0	14.0	24.0	26.0
Place Object Stand	33.0	46.0	47.0	45.0	44.0	54.0	57.0
Place Phone Stand	23.0	36.0	33.0	35.0	28.0	35.0	38.0
Place Shoe	20.0	13.0	26.0	19.0	8.0	21.0	20.0
Press Stapler	74.0	67.0	77.0	73.0	64.0	73.0	68.0
Put Bottles Dustbin	4.0	1.0	0.0	3.0	0.0	4.0	1.0
Put Object Cabinet	3.0	1.0	3.0	3.0	0.0	8.0	7.0
Rotate QRCode	19.0	8.0	22.0	15.0	4.0	16.0	9.0
Scan Object	25.0	20.0	28.0	12.0	6.0	24.0	23.0
Shake Bottle	96.0	95.0	96.0	96.0	94.0	97.0	98.0
Shake Bottle Horizontally	97.0	96.0	100.0	100.0	95.0	99.0	99.0
Stack Blocks Three	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Stack Blocks Two	8.0	0.0	6.0	7.0	2.0	7.0	6.0
Stack Bowls Three	2.0	2.0	5.0	7.0	2.0	4.0	5.0
Stack Bowls Two	51.0	43.0	53.0	63.0	10.0	62.0	59.0
Stamp Seal	9.0	2.0	6.0	5.0	1.0	8.0	5.0
Turn Switch	29.0	31.0	21.0	30.0	34.0	25.0	36.0
Average	32.74	30.52	35.82	34.80	26.24	35.22	34.40

Alternative Weighting Rules. The ACT, AAE, VOTE, and AAE + age rows of Table 4 isolate the weighting rule while holding the checkpoint, stride, and aligned candidate set fixed. For candidate a_k of age Δ_k and the freshest candidate a_f, AAE assigns $w_k \propto \exp(\alpha \cos(a_k, a_f))$ over the full

Table 14: **RoboTwin checkpoint maturity control on** π_0 (50 tasks, 100 rollouts/task), RTE at s=25.

Training step	Base	RTE	Gain
30k	33.9	42.9	+8.9
60k	38.2	48.8	+10.7

Table 15: **Per-seed breakdown of the estimator ablation on** π_0**-Long** (500 episodes per arm and seed). Mean ± population standard deviation; Δ is measured against Base.

Arm	seed 7	seed 8	seed 9	Mean	Δ
Base s=5	80.2	80.4	81.4	80.7 ± 0.5	—
Open loop s=50	65.4	64.8	69.0	66.4 ± 1.9	−14.3
Freshest only s=3	78.6	75.4	79.8	77.9 ± 1.9	−2.7
Freshest only s=1	73.4	70.6	75.2	73.1 ± 1.9	−7.6
Four samples s=5	79.2	81.6	80.0	80.3 ± 1.0	−0.4
Equal weights s=1	83.2	82.0	82.0	82.4 ± 0.6	+1.7
Equal weights s=3	79.2	83.0	83.4	81.9 ± 1.9	+1.2
Equal weights s=5	80.2	81.0	81.6	80.9 ± 0.6	+0.3
ACT (oldest first) s=3	79.0	79.2	79.4	79.2 ± 0.2	−1.5
AAE (similarity) s=3	80.4	79.0	83.8	81.1 ± 2.0	+0.4
VOTE s=3	85.6	81.2	85.0	83.9 ± 1.9	+3.3
AAE + age s=3	89.2	87.8	89.8	88.9 ± 0.8	+8.3
RTE s=1	87.2	89.6	88.8	88.5 ± 1.0	+7.9
RTE s=3	88.8	91.0	90.0	89.9 ± 0.9	+9.3
RTE s=5	90.6	89.4	90.2	90.1 ± 0.5	+9.4

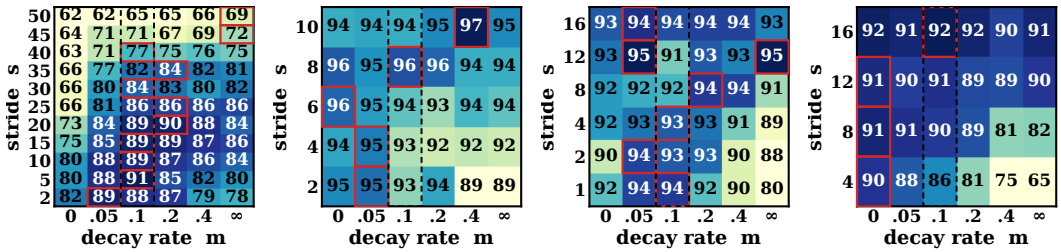
action vectors. AAE+Age retains this similarity kernel but adds the same recency factor used by RTE, $w_k \propto \exp(\alpha \cos(a_k, a_f) - m\Delta_k)$, with $\alpha = m = 0.1$. VOTE splits the candidates by whether their cosine similarity to a_f exceeds 0.5, keeps the larger group, and averages it uniformly. AAE reaches 81.1% and VOTE 83.9%, whereas AAE+Age recovers to 88.9%, only 1.0 points below RTE. The 7.9-point gap between AAE and AAE+Age identifies physical prediction age, rather than trajectory similarity alone, as the useful signal.

ACT's oldest-first rule provides the complementary directional control. We reproduce the released ACT temporal-ensembling convention, which gives the oldest aligned query the largest weight, while holding the π_0 checkpoint, the stride of $s = 3$, and the candidate set fixed. It reaches 79.2%, below both Base at 80.7% and RTE at 89.9%, and at ACT's own per-step deployment, $s = 1$, it reaches 77.1%. Same-target temporal ensembling alone therefore does not explain the gain; the preference for fresher observations is essential.

Fresh-Sample Averaging. A natural alternative to temporal aggregation is to draw several samples from the current observation and average them. With four fresh samples per query at the released stride s=5, averaging their actions and taking the median gripper command, π_0-Long reaches 80.3%, on par with Base at 80.7% and well below RTE at 90.1%, even though every query now costs four times the denoising compute. Averaging draws from a single observation therefore does not recover the headroom, which suggests that the gain of RTE comes from blending predictions made at successive observations, as the noise model of Appendix A.4 predicts.

Source-Noise Refresh. Table 16 tests whether RTE merely averages independent sampling noise. Let z_{old} be the source noise used at the preceding replan and let $\epsilon \sim \mathcal{N}(0, I)$ be a fresh draw. The next replan uses $z_{\text{new}} = \sqrt{1-\alpha}\, z_{\text{old}} + \sqrt{\alpha}\, \epsilon$, so α directly controls how much source-noise variance is refreshed. Reusing noise degrades both configurations, but RTE retains a 27.5-point advantage over Base at α=0. Here all overlapping chunks share one source draw and differ only through the updated observations, so part of the gain of RTE comes from blending successive observations rather than from averaging independent noise.

Smoothness of the Executed Motion. Blending also changes the executed motion. Table 17 measures it over the successful episodes, so that the long retries of failed episodes do not enter the comparison. The freshest-only control changes its command 2.6 times more at replanning steps than



(a) π_0 (H=50). (b) $\pi_{0.5}$ (H=10). (c) GR00T-N1.7 (H=16). (d) CogACT (H=16).

Figure 6: **Joint sweep of the decay rate m and the stride s on LIBERO-Long** for the four backbones, with strides up to the chunk length H; success rates (%) of seed 7, 500 episodes per cell. The red box marks the best rate at each stride, the dashed column m=0.1, and the column m=∞ the freshest-only control; at s=H no chunks overlap, so all rules coincide and the row shows the spread between runs.

Table 16: **Source-noise refresh on π_0-Long.** Mean $\pm$ population std over seeds 7/8/9 (500 episodes per seed). α is the fresh-noise variance fraction: α=0 reuses the previous source noise, whereas α=1 uses an independent draw. Base executes the freshest chunk at the released stride s=5, and RTE blends at s=3; bold marks the better configuration at each α.

Noise refresh α	**0**	**0.1**	**0.3**	**0.5**	**1**
Base	42.2 ± 5.4	59.0 ± 1.7	69.4 ± 1.7	73.3 ± 1.9	80.7 ± 0.5
RTE	$\mathbf{69.7 \pm 2.5}$	$\mathbf{80.7 \pm 0.4}$	$\mathbf{86.8 \pm 1.4}$	$\mathbf{88.3 \pm 2.1}$	$\mathbf{89.9 \pm 0.9}$

between them; RTE lowers this ratio to 1.4 and cuts the jerk by 58%. Smoothness alone, however, does not explain the gain. Equal weighting moves more smoothly than RTE without improving on Base, and an exponential moving average of the executed command, which smooths the motion without aligning the predictions of each step, lowers success to 73.5% with a time constant of two steps and to 7.0% at the decay rate of RTE, because it delays every command it filters. Temporal ensembling thus helps by averaging predictions of the same step issued from successive observations, not by filtering the executed trajectory.

D Task Visualizations

To provide concrete intuition for the evaluated benchmarks, we visualize frames sampled from the same camera streams consumed by the policies during successful rollouts. Following the order of the main experiments, Figure 7 illustrates LIBERO, Figure 8 presents the SIMPLER Visual Matching environments, Figure 9 showcases the diverse bimanual operations in RoboTwin 2.0, and Figure 10 details the five real-world Franka manipulation setups. Every depicted frame originates from the camera feed used by the evaluated policy. The strips are rescaled only for presentation and provide representative views of the visual observations encountered by each backbone.

E Real-World Setup and Tasks

E.1 Hardware and Environment

Our physical validation platform, depicted in Figure 3, is built around a 7-DoF Franka Research 3 manipulator equipped with a parallel-jaw gripper. Visual observations are provided exclusively by a statically mounted, top-view ZED 2i camera; all evaluated policies receive only its monocular RGB stream. State conditioning varies by backbone: π_0 and $\pi_{0.5}$ additionally consume an 8-dimensional state comprising the 3D end-effector position, a 4D end-effector quaternion, and the gripper state; GR00T-N1.7 consumes the end-effector position with a 6D rotation, the seven joint positions, and the gripper state; whereas CogACT is conditioned on the image and language instruction alone.

Table 17: **Smoothness of the executed motion on π_0-Long at s=5**, means over the successful episodes of seeds $7/8/9$. *Accel.* and *Jerk* are the mean first and second differences of the commanded end-effector translation per control step, in units of 10^{-2}, and *Boundary* is the ratio of the mean first difference at replanning steps to that at other steps. EMA filters the executed command of the freshest-only control with rate m; RTE is shaded and its success rate bold.

Configuration	Success Rate (%)	Accel.	Jerk	Boundary
Base (freshest only)	80.7	7.11	7.00	2.62
Equal weights (m=0)	80.9	4.41	1.61	1.07
RTE (m=0.1)	**90.1**	5.06	2.95	1.42
Freshest + EMA (m=0.5)	73.5	4.91	2.59	1.32
Freshest + EMA (m=0.1)	7.0	2.13	0.64	0.97

E.2 Task Definitions

We evaluate the policies across five manipulation tasks. For each task, we collect 50 expert demonstrations via 3D-mouse teleoperation at a control frequency of 25 Hz. The tasks are defined as follows (visualized in Figure 10):

- **Put Corn on Pink Plate** (*"Put the corn on the pink plate"*): The robot must place the corn so that it rests on the pink plate. This single-stage pick-and-place operation shares its object with the rack task but targets a flat receptacle on the table.
- **Press Yellow Button then Blue** (*"Press the yellow button, then press the blue button"*): The robot must press the yellow button and then the blue button, each once and in this order. The task involves no grasp but requires two precise contacts in sequence.
- **Cover Blue Block with the Box** (*"Cover the blue block with the box"*): The robot must lower the box over the blue block until the block is fully covered. This is the most demanding spatial-alignment task in the suite because the box must be centered while occluding its target during placement.
- **Lift Corn onto Upper Rack Shelf** (*"Put the corn on the upper rack"*): The robot must place the corn so that it rests on the upper shelf of the rack. The placement requires precise vertical clearance to pass the shelf threshold.
- **Move Tomato from Turntable to Plate** (*"Pick up the tomato from the rotating turntable and place it in the white plate"*): The robot must grasp the tomato from the rotating turntable and place it on the white plate. Because the turntable keeps rotating, the grasp target moves throughout the approach.

E.3 Deployment Protocol and Control Deadlines

During deployment, every chunk is converted into absolute end-effector targets with an absolute gripper command, so all three deployments execute, and RTE blends, predictions in the same coordinates, and the robot tracks these targets in Cartesian space at 25 Hz. Each configuration is evaluated over 40 trials per task. Object layouts and initial poses are randomized, but each task's layout sequence is generated in advance and shared by Base, the freshest-only control, and RTE, which run on every layout in rotating order. A trial is deemed successful only if all stages of the language instruction are completed within the time limit; no partial credit is awarded for multi-stage tasks.

Hardware-Constrained Strides. At 25 Hz, the robot controller consumes one action every 40 ms, whereas a π_0-family forward pass takes approximately 72.5 ms on this accelerator class, as measured in Table 8. Non-blocking per-step replanning is therefore infeasible on this platform. The freshest-only control and RTE replan every 10 steps for π_0, $\pi_{0.5}$, and GR00T-N1.7 and every 8 steps for CogACT, which leaves ample time for inference, image capture, and communication, while Base executes each chunk open-loop. Under $m = 0.1$, these strides keep the ESS at 2.1 for $H = 50$ and $H = 40$ and at 1.7 for $H = 16$, the range in which blending helps on RoboTwin. RTE then averages at most $\lceil H/s \rceil = 5$ candidate actions per control step, and this weighted average over low-dimensional action vectors adds no measurable latency relative to a policy forward pass.



Figure 7: **Two LIBERO instructions from each of the four suites**, eight frames apiece, sampled uniformly from one successful π_0 rollout in the agent-view camera that the policy consumes.

F Limitations and Future Scope

While RTE provides a lightweight and training-free mechanism for recovering stochastic headroom, several operational boundaries offer natural directions for future work. First, our pass@N diagnostic relies on repeated environment resets, leaving online estimation of headroom within a single closed-loop rollout an open problem. Second, while simulation encompasses 50 bimanual tasks, our physical validation is currently conducted on a single Franka arm; testing whether recency weighting

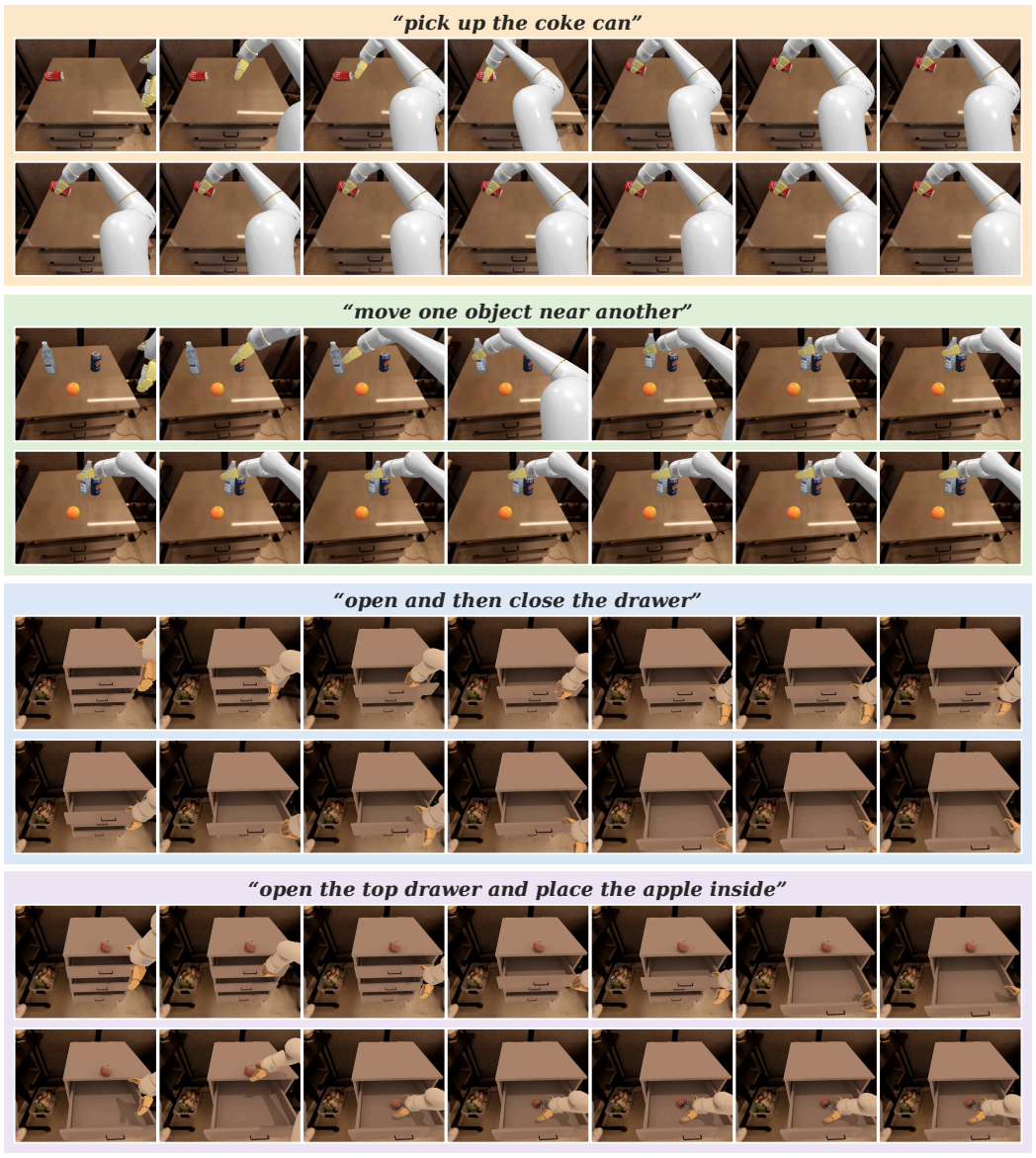


Figure 8: **The four SIMPLER Visual Matching instructions**, fourteen frames apiece from one successful CogACT rollout in the Google robot's static camera, read left to right over two rows. Both drawer tasks render under ray tracing, as the released protocol specifies.

preserves coordination on physical dual-arm or humanoid platforms is an important next step. Finally, dynamically adapting the decay rate m online based on instantaneous plan disagreement represents a natural progression beyond a fixed global constant.



Figure 9: **Ten RoboTwin 2.0 instructions**, eight frames apiece, sampled uniformly from one successful $\pi_{0.5}$ rollout at s=10 in the `head_camera` view that the policy consumes.

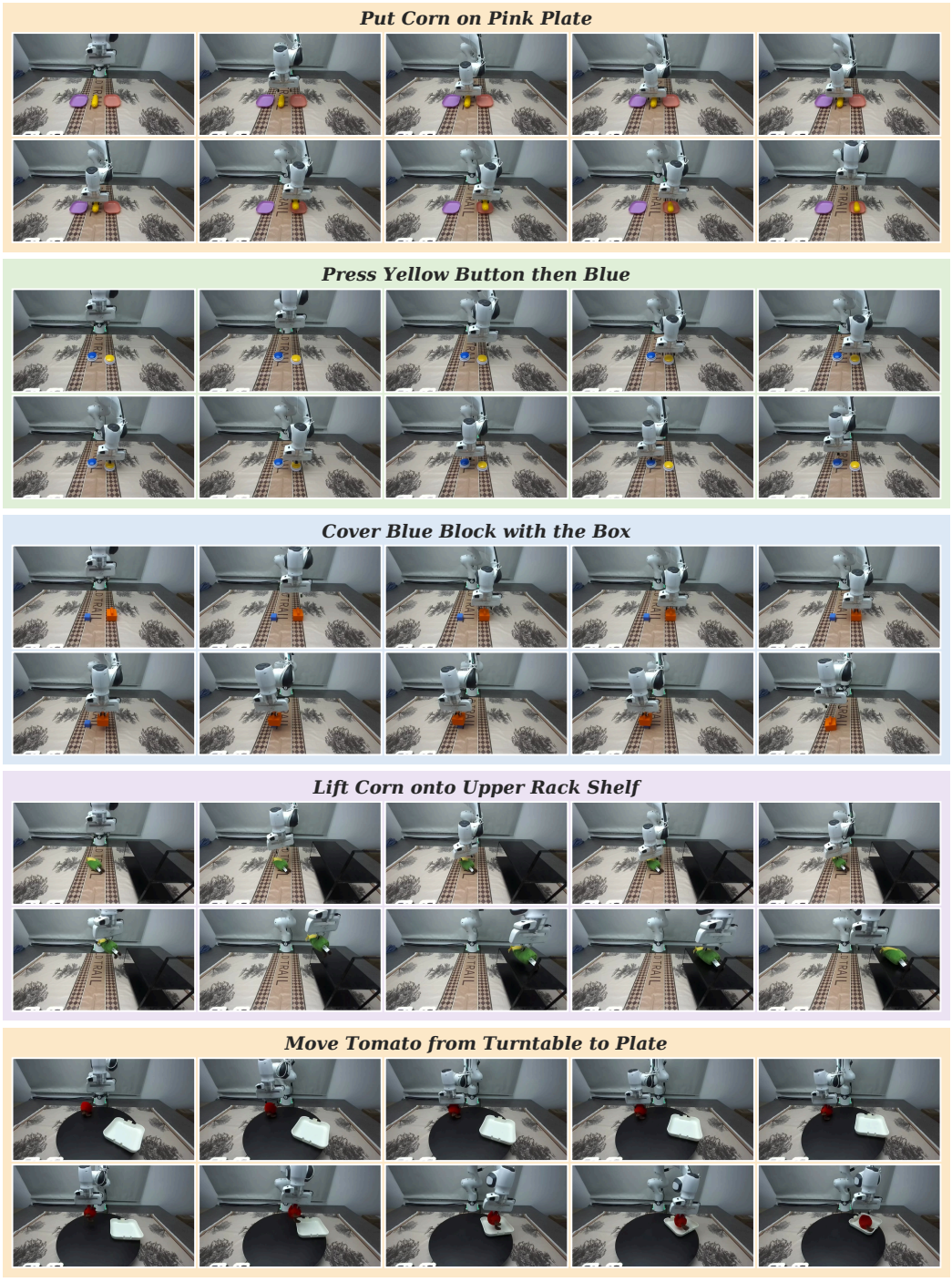


Figure 10: **The five Franka Research 3 tasks**, ten frames apiece from one teleoperated demonstration, read left to right over two rows, in the uncropped ZED 2i top-camera view that the evaluated policies receive.