STREAM-IN-STREAM: TEACHING VISION-LANGUAGE-ACTION MODELS WHEN THEIR ACTIONS EXPIRE

Anonymous authors
Paper under double-blind review

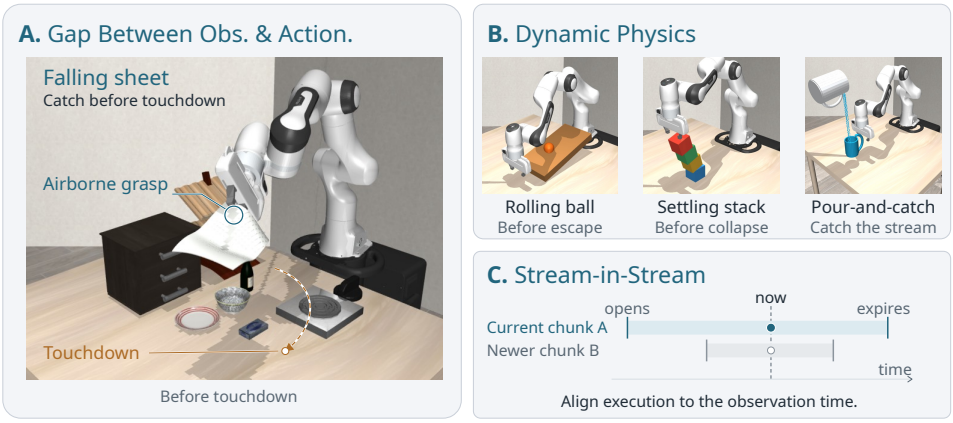

Figure 1: **The world keeps moving while a policy computes.** (A) A falling sheet must be grasped before touchdown, but an action computed from an older frame arrives after the scene has moved. (B) Other families with finite opportunities to act. (C) A validity window holds the start times from which a chunk, executed time-aligned to its observation, still succeeds; the newer chunk B's window ends first, so the scheduler keeps A.

ABSTRACT

Vision-language-action (VLA) models are mostly trained and evaluated in scenes that hold still while they compute. We study the stream-in-stream setting, in which the world keeps moving during inference, and make action timing measurable and learnable. PhysClock adds granular, settling, rolling, and air-coupled task families to simulated manipulation and advances the world while a policy computes; PhysClock-Syn labels scripted demonstrations with replay-measured timing: how much delay each decisive action tolerates, and how early and how late each action chunk may start. Fine-tuned on these demonstrations and run synchronously, $\pi_{0.5}$ still gains 14 percentage points when the world pauses during its inference. We train a small head on the timing labels to predict the earliest and latest start of each chunk, and a scheduler waits for, keeps, or discards chunks accordingly. In a confirmatory evaluation with contrasts fixed in advance, the head raises success on the three families other than a granular pour by 3.6 and 5.5 points on two policies, mostly in episodes with little timing tolerance, also when attached to frozen policies; a head that predicts success probabilities from the same labels is equivalent to it within a pre-specified margin. On a real arm, heads trained on event-time proxies from demonstration videos raise grasps of a falling sheet from 58 to 83 of 200 paired trials in a pre-registered batch, after a smaller one was inconclusive, and catches of a rolling ball from 41 to 60 of 120. A head trained in simulation does not transfer to the real arm, and on a task family held out from training the head brings no measurable gain.

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

1 INTRODUCTION

A robot that catches a granular pour in a cup, grasps a falling sheet, or lifts a block from a stack that is still wobbling acts in a world with its own clock. By the time an action reaches the motors, it was computed for a scene that has moved on, and it is useful only if the state it was meant for has not yet passed. Delayed MDPs and real-time reinforcement learning formalize this setting (Katsikopoulos & Engelbrecht, 2003; Ramstedt & Pal, 2019). We call it *stream-in-stream*: the stream of a policy's inference calls runs inside the physical stream of the world, and each output is useful only during a window of the outer stream.

Vision-language-action (VLA) models (Kim et al., 2024; Black et al., 2025a;b; Pertsch et al., 2025; NVIDIA et al., 2025) are mostly trained and evaluated in the opposite setting. Simulators step the world only when an action returns (Liu et al., 2023; Zhu et al., 2020); action chunking (Black et al., 2025b; Kim et al., 2025) commits to tens of future steps from one observation; and success is scored without regard to when the decisive action landed. Recent benchmarks let the world move while a policy computes (Xie et al., 2026; Chen et al., 2026b; Liao et al., 2026; Liu et al., 2026a), but they use rigid objects, and none measures how much delay each action tolerates.

We first make timing measurable (§3). PhysClock advances the world during inference by the latency a policy incurs on a reference accelerator, replayed from recorded profiles, and PhysClock-Syn labels every demonstration with the replay-measured *timing tolerance* of its decisive action. Fine-tuned on PhysClock-Syn and run synchronously, $\pi_{0.5}$ gains 14 points when the world pauses during its inference (§4), as other policies do on rigid dynamic tasks (Sun et al., 2026; Liu et al., 2026a): latency, not only skill, limits it.

We then use the labels to choose among action chunks (§5). In plain terms, a small head predicts for every chunk how early and how late it may start and still succeed, its *validity window*, and a scheduler waits for a window to open, keeps the active chunk unless a newer chunk's window ends later, and discards chunks whose windows have closed. The head is trained on labels from simulator replays, from event-time proxies derived from demonstration videos, or from the policy's own outcomes, and attaches to any chunked policy. We test it on OpenVLA-OFT, on A2C2 (Sendai et al., 2025), and on a dual-rate $\pi_{0.5}$ policy (DR-$\pi_{0.5}$) in the style of HiRT and πR^2 (Zhang et al., 2024; Park & Tulsiani, 2026) with a short-horizon motion encoder.

In a confirmatory evaluation with contrasts fixed in advance (§6.2), the head raises success on the three families other than the pour by 3.6 to 5.5 points on two backbones, mostly in timing-critical episodes, and leaves the pour unchanged. Attached with a stop-gradient to frozen policies it does as well, and a value head trained on the same labels is equivalent to it within a pre-specified margin (§6.3), so the gain comes from the timing labels and a learned choice among chunks. With heads on both, DR-$\pi_{0.5}$ cannot be distinguished from a delay-conditioned OpenVLA-OFT (§6.4). On a real arm, heads trained on event-time proxies raise success on a falling sheet and a rolling ball, and on the sheet a tuned constant window does not match them (§6.5). Our contributions are (i) PhysClock, which adds granular, settling, and air-coupled dynamics with matched quasi-static and paused counterparts, and PhysClock-Syn, whose replay-measured timing labels make action timing measurable; (ii) a measurement of what latency costs a VLA fine-tuned on dynamic demonstrations; (iii) two-sided validity windows learned from these labels and used to arbitrate among chunks computed from different observations, with three label sources and an equivalence test against value-based selection; and (iv) an evaluation protocol with shared latency accounting, a confirmatory split, and pre-registered paired real-robot trials.

2 RELATED WORK

Latency-aware execution and reactive VLAs. Action chunking (Black et al., 2025b; Kim et al., 2025; Pertsch et al., 2025) amortizes inference but commits to a chunk computed from one increasingly stale observation; temporal ensembling (Zhao et al., 2023) and closed-loop resampling (Liu et al., 2025) re-decide the executed action at every step. RTC (Black et al., 2025c;d) repairs the seam between chunks, and other methods react between chunks with per-step corrective heads (Sendai et al., 2025; Agouzoul, 2026), event-triggered re-inference (Wu et al., 2026), multi-frame context (Liu et al., 2026b), or dual-system policies (Zhang et al., 2024; NVIDIA et al., 2025; Chen et al., 2025; Park & Tulsiani, 2026; Song et al., 2025), some trained on a stale prefix as ours (Yan

et al., 2026; Sun et al., 2026; Li et al., 2026b; Figure AI, 2025). For moving objects, DynamicVLA discards steps outdated by the measured delay (Xie et al., 2026), and others predict execution-time features or regenerate chunks from motion history (Syed et al., 2026; Chang & Deng, 2026). Closest to our head, learned execution horizons decide how long to run a chunk before re-planning (Zhao et al., 2026; Xu et al., 2026; Lei et al., 2026), learned selectors pick among chunks predicted at different times (Weng et al., 2025) or among the candidates of one asynchronous call with a chunk-level critic (Dong et al., 2026), and value-based rules keep a committed plan unless a new one is better (Zhang et al., 2022; So et al., 2025). These run synchronously, bound only how long a chunk may run, or compare candidates of one observation; our head predicts a two-sided activation window from delay-indexed replay labels and arbitrates among chunks computed from different observations, reusing time-aligned execution (Chi et al., 2024) and asynchronous inference.

Acting under delay and streaming perception. Delayed MDPs (Katsikopoulos & Engelbrecht, 2003) and real-time and delayed reinforcement learning (Ramstedt & Pal, 2019; Bouteiller et al., 2021; Derman et al., 2021; Wang et al., 2024a) augment the state with the pending actions or predict the state at execution time (Walsh et al., 2009; Firoiu et al., 2018; Chen et al., 2021; Kim et al., 2023; Karamzade et al., 2024); concurrent and asynchronous control keeps acting while computing (Xiao et al., 2020; Travnik et al., 2018; Yuan & Mahmood, 2022; Riemer et al., 2025), and sequence-based anytime control (Quevedo & Gupta, 2013) buffers future inputs and applies the entry due now, the closest analogue of a chunk buffer with time-aligned execution. In this view a chunk's validity window is a superlevel set of the value of activating it in the augmented state; our head learns a summary of that value from replays, a value head fit on the same replays is its value-based counterpart, and the conditioned OpenVLA-OFT of §6.1 is the augmented-state baseline trained by imitation (Appendix F). TempoRL learns how long to commit to an action (Biedenkapp et al., 2021), and real-time systems attach validity intervals and time-utility functions to computed values (Ramamritham, 1993; Jensen et al., 1985). Streaming perception scores each output against the world when it becomes available and simulates the clock from runtimes on a reference GPU (Li et al., 2020), as we do; observation age is an instance of the age of information (Kaul et al., 2012).

Dynamic benchmarks and simulation. Recent benchmarks let the world move while a policy computes, converting measured latency into simulated time (Xie et al., 2026; Chen et al., 2026b; Liao et al., 2026; Liu et al., 2026a; Wang et al., 2026), and pausing the world exposes what latency costs (Sun et al., 2026; Liu et al., 2026a); TEMPO instead attributes dynamic failures to missing temporal context (Feng et al., 2026). These benchmarks use rigid objects. Pouring has stayed near quasi-static (Schenck & Fox, 2017; Lin et al., 2023; Chen et al., 2026a), airflow is exploited on hardware because it is hard to simulate (Xu et al., 2022; Bhardwaj et al., 2024), and particle and shell simulators lack air or model it offline (Lin et al., 2020; Gu et al., 2023; Wang et al., 2024b; Zhu et al., 2026; Xian et al., 2023). PhysClock adds granular, settling, and air-coupled families with matched quasi-static and paused counterparts, and PhysClock-Syn adds what none of these benchmarks measures, the timing tolerance of each decisive action. Options (Sutton et al., 1999) and robot catching are discussed in Appendix F.

3 Problem and Benchmark

3.1 Timing, validity, and scheduling

Problem. A chunked policy reads an observation at time t_o and returns, after a latency ℓ, a chunk $a = (a_0, \ldots, a_{k-1})$ whose step a_i falls due at $t_o + i/f_c$ for the control rate f_c. At every control tick t a scheduler holds the active chunk $c^\star$, activated at $\tau^\star$, and a buffer of returned candidates; it continues $c^\star$, activates a candidate, or holds the pose. It sees the observations and the policy's outputs, not the future of the world, and its objective is the probability that the episode succeeds; when the policy is called is fixed by its loops and lies outside the problem.

Activation. Each step carries an end-effector pose delta and a gripper command, open or closed. Activating a at a tick $\tau \geq t_o + \ell$ issues, at each tick $t \geq \tau$ until a is replaced or runs out, the step due at t, $a_{\lfloor (t-t_o) f_c \rfloor}$ (time-aligned execution). The deltas of steps that fell due before τ are dropped, so the chunk's remaining motion starts from the pose in which the preceding behavior left the arm, and its gripper level takes effect at τ.

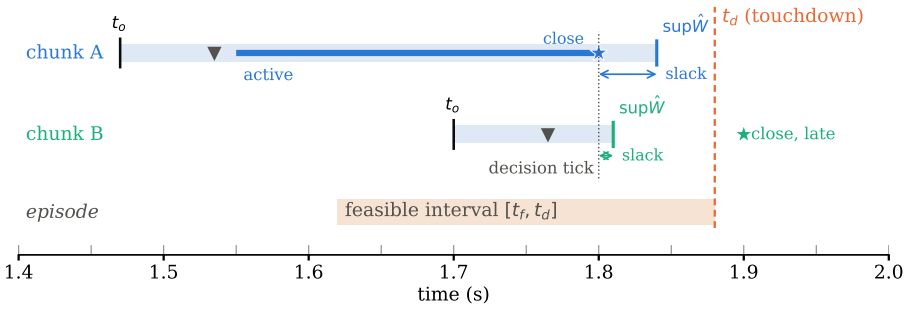

Figure 2: Timing notions on one F4 episode. Each chunk is computed from the frame at t_o and becomes available at the triangle; the bar is its predicted window $\hat{W}$, the star its gripper close. At the decision tick B's window ends before A's, so the scheduler keeps A, which closes inside the feasible interval; B's window is optimistic, since B closes after touchdown, but the order suffices. Executing the newest chunk would switch to B.

Table 1: Notions and abbreviations: the first three rows concern a chunk, the others an episode, a regime, a metric, or a method.

Value V; window W^α	success probability of activating a at τ (Eq. 1); activation times with $V \geq 1-\alpha$
Realized window; ϵ, $\delta_{\max}$	first interval of successful activation times in one realization; its ends relative to t_o
Predicted window $\hat{W}$; slack	$[t_o + \hat{\epsilon},\ t_o + \hat{\delta}]$ from the window head; slack at tick t: $t_o + \hat{\delta} - t$
Decisive action; $[t_f, t_d]$	action that decides the episode; its feasible interval, from the enabling event to the deadline
Tolerance; TC episodes	shift interval of the expert's decisive segment with success; timing-critical: tolerance below 400 ms
QS, DYN, PW	quasi-static counterpart; physical stream, world advances during compute; paused world during compute
On-time rate (OTR); lateness; age; hold	decisive action inside $[t_f, t_d]$; its time minus t_d; time from the newest frame behind an executed step to its issue; share of control time without a new step
s/s; RTC; A2C2; OFT	accelerator-seconds per second of control; real-time chunking (Black et al., 2025c); residual correction (Sendai et al., 2025); OpenVLA-OFT (Kim et al., 2025)

Value and windows. Let β be the behavior of the robot on $[t_o, \tau)$ and π the policy that acts once a runs out. The success probability of activating a at τ given the history h_{t_o} is

$$V_{\pi,\beta}(a, \tau; h_{t_o}) = \Pr\big(\text{success} \mid h_{t_o},\ \beta \text{ on } [t_o, \tau),\ a \text{ activated at } \tau,\ \text{then } \pi\big), \tag{1}$$

and the level-α *validity window* is $W^\alpha = \{\tau \geq t_o + \ell : V_{\pi,\beta} \geq 1 - \alpha\}$. In one realization of the world's randomness the outcome is deterministic, and the first maximal interval of successful activation times, $[t_o + \epsilon,\ t_o + \delta_{\max}]$, defines the *earliest start* ϵ and the *maximum admissible delay* $\delta_{\max}$ that replays measure. Our replay labels fix β as holding the pose and π as the scripted expert; at deployment β is the previously active chunk and π the policy, and relabeling with these moves $\delta_{\max}$ by a median of $+11$ ms for β and -52 ms for π (Appendix A).

Episodes. Each task family defines a *decisive action*, such as closing the gripper on the target, that must land in the episode's *feasible interval* $[t_f, t_d]$, from the event that makes it possible to the deadline. We consider tasks with one decisive action per episode; several decisive actions, or timing constraints that last through a task, are outside our scope. The *timing tolerance* of an episode is the length of the interval of time shifts of the scripted expert's decisive segment under which the expert still succeeds (§3.3); it is the expert's, fixed before any policy acts, not the tolerance an evaluated policy faces. Table 1 collects the notions used below, and Figure 2 shows them on one episode.

3.2 PhysClock

PhysClock is built on LIBERO (Liu et al., 2023), robosuite (Zhu et al., 2020), and MuJoCo (Todorov et al., 2012) with the conventions of the $\pi_{0.5}$ LIBERO recipe. Pours are granular, 300 to 1,200 rigid spheres in training; sheets are shell elements coupled through a drag closure to a coarse-grid air solver with gusts, with the size, mass, and stiffness of the real sheet of §6.5, whose fall time and descent near the table agree with 40 filmed real drops (Appendix D); stacks and balls are rigid bodies with randomized impulses and releases. Table 2 lists the four families, each with five instances, randomized onsets, poses, and physical parameters, and a QS counterpart. The world clock runs independently of compute: in DYN, a call issued at t reads the newest frame and returns at $t + \ell$, with ℓ replayed from a profile recorded on one A800 or injected as a constant (Appendix A).

Table 2: Task families and dataset statistics. Tolerance: median length (ms) of the interval of time shifts under which the expert's decisive action succeeds, on the $\pm$400 ms grid of the labels, censored at 800 ms ($\geq$: median of lower bounds), and on a $\pm$1,000 ms relabeling; lower b.: fraction of $\pm$400 ms tolerances that reach an end of the grid; last column: fraction of tolerances below one $\pi_{0.5}$ inference (190 ms).

Family	Decisive action	Deadline event	Demos	Expert SR	Tol. $\pm$400	Tol. $\pm$1,000	Lower b.	$<$190 ms
F1 Pour-and-Catch	cup under the stream	pour ends or spill limit	2,500	97.3%	$\geq$715	1,020	98%	0.0%
F2 Settling Stack	grasp target block	block falls, stack collapses	2,500	94.6%	$\geq$626	700	59%	1.3%
F3 Rolling Ball	intercept and grasp	ball leaves reach	2,500	94.4%	380	380	26%	4.2%
F4 Falling Sheet	grasp in the air	sheet touches table	2,500	89.6%	280	280	8%	19.4%

Table 3: Left: QS-to-DYN degradation after fine-tuning on quasi-static demonstrations (SR and OTR in %, 1,200 episodes, expert 400; lateness and age in ms). Right: success in % of $\pi_{0.5}$ under interventions with fixed weights (1,200 episodes, training run 1), fine-tuned on quasi-static demonstrations or on PhysClock-Syn; the QS counterparts of F1 and F4 are different tasks (Appendix A).

Model	QS	DYN	OTR	Late	Age
$\pi_{0.5}$	71.8	14.8	33.0	156	433
π_0-FAST	60.7	7.2	16.1	402	647
OpenVLA-OFT	62.5	24.0	54.7	$-$42	272
SmolVLA	54.2	22.0	48.3	7	229
Scripted expert	96.0	93.5	99.8	$-$333	–

Intervention	QS-tuned	PhysClock-Syn-tuned
None (DYN)	14.8	43.6
Paused world (PW)	24.4	58.0
PW, chunk k=5	39.1	59.6
QS counterpart	71.8	68.0

3.3 PHYSCLOCK-SYN: DEMONSTRATIONS WITH TIMING LABELS

A scripted expert per family reads ground-truth state, incurs no latency, and looks ahead by rolling the simulator forward. *Tolerance labels.* The decisive segment of a demonstration runs from the expert's last replanning before the decisive action, 150 to 300 ms earlier, to 200 ms after it. We replay each episode under its seed with this segment shifted by $\Delta \in [-400, 400]$ ms in 50 ms steps, keeping the earlier segments fixed, holding the pose in the gap that a positive shift opens, and starting the later segments when the shifted one ends; the maximal successful interval containing $\Delta = 0$, censored at the 800 ms width of the grid, gives the tolerance, with ends refined by bisection to 12.5 ms. *Window labels.* Chunks sampled every 100 ms within 1 s of the decisive action are replayed after holding for 0, 50, 100, 200, and 400 ms, activated time-aligned as in Eq. 1 and continued by the expert, with two bisections at the right bracketing gap; the smallest successful delay gives ϵ and the last success before the next sampled failure gives $\delta_{\max}$, censored at 400 ms (Appendix A). For 6% of decisive actions (19% in Falling Sheet) the expert's own timing tolerance is shorter than one forward pass of the backbone (Table 2), although the feasible intervals last longer (366 ms on average in F4). Re-grasps by the expert rescue 3.9% of the successful shifted replays; without them the timing-critical head effects move by at most 0.2 points (Appendix A).

4 WHAT LIMITS VLAS IN PHYSICAL STREAMS

Degradation. We fine-tune four open VLAs with different action heads, $\pi_{0.5}$ (Black et al., 2025a) (flow matching), π_0-FAST (Pertsch et al., 2025) (discrete tokens), OpenVLA-OFT (Kim et al., 2025) (parallel decoding), and SmolVLA (Shukor et al., 2025) (asynchronous inference stack), on expert demonstrations of the QS counterparts with identical budgets. All four lose 32 to 57 points in DYN (Table 3, left); $\pi_{0.5}$ falls from 71.8% to 14.8%, and its decisive action lands a median 156 ms late. These models never saw a moving scene, so the drop mixes the regime with a change of task.

What limits a policy that can intercept. Fine-tuned on PhysClock-Syn, $\pi_{0.5}$ reaches 43.6% in DYN. Pausing the world during its inference, with weights fixed, raises it by 14.4 points to 58.0%, and by 15.5 and 14.5 points on two further training runs, whereas halving the executed chunk as well adds only 1.6 points (Table 3, right): latency, rather than action granularity, is the larger remaining cost. For the model fine-tuned on quasi-static demonstrations the order reverses: pausing adds 9.6 points, halving the chunk 14.7 more, and it still stays 32.8 points below its QS level, since it never learned to intercept (Appendix C).

Algorithm 1 Validity-gated scheduler (control loop at rate f_c)

Require: candidate buffer B filled asynchronously by the action loop with $(a, \hat{W}, t_o)$ when a call returns, so only $\hat{W} \cap [t_o + \ell, \infty)$ is used; a candidate c is *expired* at t if $t > \sup c.\hat{W}$ or $c.a$ has no step left for t, and a candidate with an empty window ($\hat{\epsilon} \geq \hat{\delta}$) is never selected; $\text{slack}(c, t) = \sup c.\hat{W} - t$

$c^\star \leftarrow \emptyset$
for each control tick at time t **do**
  discard all expired candidates from B; **if** $c^\star$ is expired, $c^\star \leftarrow \emptyset$ (safeguard; -0.2 points if removed)
  $c \leftarrow$ most recent candidate in B with $t \in c.\hat{W}$
  if $c \neq \emptyset$ **and** ($c^\star = \emptyset$ **or** $\text{slack}(c, t) > \text{slack}(c^\star, t)$) **then**
    $c^\star \leftarrow c$
  end if
  if $c^\star \neq \emptyset$ **then**
    execute the step of $c^\star.a$ indexed by $\lfloor (t - c^\star.t_o)\, f_c \rfloor$
  else
    hold the pose; signal the action loop to re-plan now
  end if
end for

5 Validity Windows

The diagnosis asks for a policy that knows how long each decision stays valid: a small head rates every proposed chunk with the earliest and latest time at which it may start and still succeed, and a scheduler acts on these windows at every control tick (Fig. 1C and Fig. 3).

Window head. A two-layer MLP reads the policy's pooled action-decoder states for a chunk (for $\pi_{0.5}$, from one extra pass of its flow-matching expert on the finished chunk) and predicts $\hat{\epsilon}$ and $\hat{\delta}$, defining $\hat{W} = [t_o + \hat{\epsilon},\ t_o + \hat{\delta}]$. It is trained jointly with the policy, or with a stop-gradient on frozen weights, by a Huber loss on $\log(x + 50\,\text{ms})$ with weight 0.1, which approximates median regression: the head estimates the conditional medians of the window ends given its inputs, under the behavior the labels assume and over the chunks it is trained on. Pinball losses at levels $1 - \alpha/2$ for $\hat{\epsilon}$ and $\alpha/2$ for $\hat{\delta}$ estimate quantiles instead; were these the exact conditional quantiles given the head's inputs, and the successful activation times of every realization an interval, the predicted interval would lie inside W^α conditional on those inputs, by a union bound (Appendix B). Training labels come from demonstration chunks and from *timing-perturbed copies* whose decisive segment is moved by up to ± 300 ms, which enter only the window loss and teach $\hat{\epsilon} > 0$, since a premature copy succeeds only when activated later. *Replay labels* (§3.3) hold the pose on $[t_o, \tau)$ and continue with the scripted expert. *Event-time proxies*, $\tilde{\epsilon} = \max(0, t_f - t_o - d)$ and $\tilde{\delta} = t_d - t_o - d$, use the offset d of the decisive step and event times from simulation or video. They approximate a translated chunk, not the exact time-aligned replay window (Appendix A). *Outcome labels* bound ϵ or δ by the timely, late, or early outcome of each chunk in the policy's own rollouts, with β the previously active chunk and π the policy.

Scheduler. Algorithm 1 discards expired candidates, lets the newest valid candidate replace the active chunk if its predicted window ends later, and holds when none is valid. It is a heuristic for value-based option interruption (Sutton et al., 1999). An activation window does not define the value of continuing an active chunk: reusing its end as an expiry threshold is a safeguard, not a consequence of the window definition. Later window ends also serve as a ranking score, tested empirically in §6.3. Comparing ends gives 66.1%; crediting the active chunk with its slack at activation gives 65.8% (Appendix D).

DR-$\pi_{0.5}$. We also pair the head with a dual-rate $\pi_{0.5}$ policy (Fig. 4). As in HiRT and πR^2, a prefix loop refreshes the vision-language key-value cache at $f_v = 3$ Hz and an action loop reruns the action expert and the head at about 15 Hz on the latest cache, so the backbone is off the action path; a motion loop encodes the last 8 frames of both cameras into 16 tokens z_t at 30 Hz. *Stale-prefix*

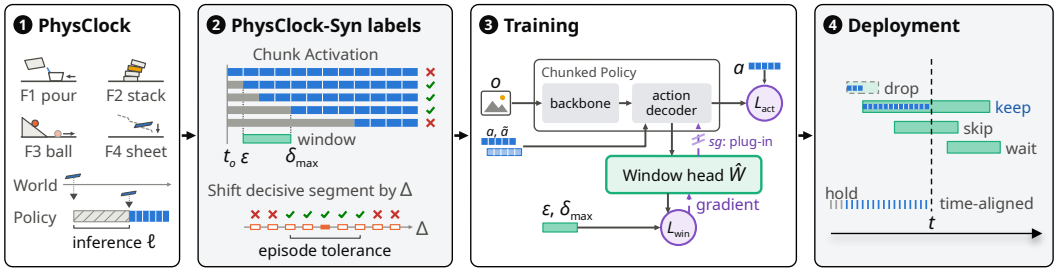

Figure 3: Overview. ❶ PhysClock advances the world during inference. ❷ Delayed replays label each chunk's earliest and latest start (ϵ, $\delta_{\max}$); shifted replays label each episode's tolerance. ❸ A window head on any chunked policy learns $\hat{W}$ beside the action loss, or as a plug-in behind a stop-gradient (sg). ❹ Algorithm 1 drops, waits for, keeps, or skips candidates by their windows.

training takes the prefix image $U(0, 300)$ ms earlier than the rest of the input, so that the expert reads the current state from z_t (Appendix B).

6 Experiments

6.1 Setup

Baselines. Standard VLAs fine-tuned on PhysClock-Syn without its labels ($\pi_{0.5}$, OpenVLA-OFT; π_0-FAST and SmolVLA in Appendix C); latency-aware variants of $\pi_{0.5}$: RTC (Black et al., 2025c) with its execution horizon equal to the inference delay and latency augmentation, A2C2 (Sendai et al., 2025) with our 8-frame encoder features, and continuous inference with DynamicVLA's execution; DynamicVLA itself (Xie et al., 2026) (46.2% in development from its own, weaker checkpoint); and OpenVLA-OFT re-planning at every tick with delay augmentation, also conditioned on its expected delay and pending actions, the augmented-state baseline of delayed RL trained by imitation, which we call the conditioned OpenVLA-OFT (Appendix B). All methods share demonstrations, backbone training steps, control rate, and cameras; controls are tuned on validation initial conditions (Appendix B), and only DR-$\pi_{0.5}$'s encoder has a privileged loss. *Development evaluation.* 400 initial conditions under three seeds (1,200 episodes) per training run, over three runs unless a table states one; each call is charged its own preprocessing and transport. We treat its results as exploratory. *Confirmatory evaluation.* It uses 2,000 fresh initial conditions under three seeds (6,000 episodes per run) and the protocol we propose for PhysClock: a shared frame buffer, profiles recorded under deployment concurrency, sensing and actuation delays, and offsets drawn per episode. Every model was trained anew, with training seeds 4 to 6, after the plan was fixed, and no run was excluded. The plan names success on F2 to F4 as the primary endpoint, since 98% of F1's tolerances are lower bounds, and two primary contrasts, the head's effect on the conditioned OpenVLA-OFT and on DR-$\pi_{0.5}$, each against the same policy retrained without the head, Holm-corrected; the head's effect on two further configurations, timing-critical success, F1 as a negative control with a $\pm$2-point equivalence margin, and comparisons across systems are secondary. An amendment fixed before any plug-in head was trained adds stop-gradient window and value heads on the headless weights, with a $\pm$2-point margin for window minus value. *Statistics.* Intervals over three runs add the between-run variance of paired per-run differences to the episode-level variance, with Satterthwaite t quantiles; real-robot pairs use exact McNemar tests. Differences are computed before rounding; points are percentage points. Table 7 maps each claim to its evidence.

6.2 Confirmatory evaluation

In development runs (Table 4), continuous re-planning helps most among the baselines, and the two policies with a window head lead. The confirmatory evaluation tests the head directly (Table 5). On the primary endpoint the head adds 3.6 points [2.2, 5.0] to the conditioned OpenVLA-OFT and 5.5 [4.0, 7.0] to DR-$\pi_{0.5}$, and 4.0 points to the delay-augmented OpenVLA-OFT and 3.9 to A2C2; Holm-corrected over the four, every p is below 0.001. The effect is concentrated where timing is tight: on timing-critical episodes the head adds 5.9 to 8.7 points, 4.8 [2.3, 7.4] and 6.7 [4.1, 9.3]

Table 4: Development evaluation (exploratory), DYN, training run 1: success in % per family, overall, and on the 473 timing-critical episodes (TC); 1,200 episodes per row, the expert 400 on seed 1 with TC over its 144 timing-critical episodes; s/s: accelerator-seconds per second of control; "+ head": window head and Algorithm 1. Three-run means for most rows are in Appendix C.

Method	F1	F2	F3	F4	All	TC	s/s
$\pi_{0.5}$, synchronous	79.0	46.3	39.0	10.0	43.6	16.5	0.28
OpenVLA-OFT, synchronous	77.0	68.0	55.0	22.0	55.5	35.3	0.17
$\pi_{0.5}$ + RTC ($s{=}d$, latency augmentation)	84.3	66.7	47.3	17.7	54.0	27.3	0.95
$\pi_{0.5}$ + A2C2 (8-frame encoder features)	83.3	70.7	59.3	30.7	61.0	41.6	1.30
$\pi_{0.5}$ + lat. aug., back-to-back (DynamicVLA's execution)	83.3	63.3	54.0	19.7	55.1	34.5	1.00
OpenVLA-OFT, conditioned	80.3	74.3	62.0	30.3	61.8	42.5	1.61
OpenVLA-OFT, conditioned + head	82.3	77.3	67.3	34.0	65.3	48.4	1.62
DR-$\pi_{0.5}$ + head	86.7	71.3	65.3	37.3	65.2	50.5	1.42
Look-ahead expert	98.0	90.0	95.0	91.0	93.5	89.6	–

Table 5: Main result: confirmatory evaluation (three training runs per model, 6,000 episodes per run). Success in % on F2 to F4 (primary endpoint, 4,500 episodes per run), on timing-critical episodes (TC, 2,368 per run), and on F1 (1,500), without and with a jointly trained window head; head effect with 95% intervals (90% for F1, the negative control). †Primary contrasts, Holm-adjusted $p < 0.001$. Plug-in heads are in Table 6, per-run values in Table 11.

	Without head			With head			Head effect		
Policy	F2–F4	TC	F1	F2–F4	TC	F1	F2–F4	TC	F1
OpenVLA-OFT, conditioned†	50.0	35.3	80.6	53.6	41.2	80.4	3.6 [2.2, 5.0]	5.9 [4.0, 7.8]	−0.2 [−1.9, 1.5]
OpenVLA-OFT, delay aug. only	49.1	34.6	80.5	53.1	40.9	80.6	4.0 [2.6, 5.4]	6.3 [4.4, 8.1]	0.1 [−1.6, 1.8]
A2C2 + 8-frame encoder features	48.6	35.3	81.7	52.5	41.2	81.7	3.9 [2.5, 5.3]	5.9 [4.0, 7.8]	0.0 [−1.6, 1.6]
DR-$\pi_{0.5}$†	47.4	32.9	84.0	52.9	41.6	83.8	5.5 [4.0, 7.0]	8.7 [6.9, 10.5]	−0.2 [−1.7, 1.3]
$\pi_{0.5}$, single loop + 8-frame tokens	–	–	–	49.6	38.6	83.0			
$\pi_{0.5}$, synchronous	29.4	14.9	76.2	–	–	–			

points more than on the other episodes of F2 to F4 for the two primary contrasts, and on F1 all four effects are equivalent to zero within ±2 points. With heads on both, DR-$\pi_{0.5}$ and the conditioned OpenVLA-OFT cannot be distinguished (−0.7 [−2.6, 1.2] points on F2 to F4); within the $\pi_{0.5}$ backbone, DR-$\pi_{0.5}$ leads single-loop $\pi_{0.5}$ with the same 8-frame tokens and head by 3.3 [1.5, 5.1].

6.3 What the window learns

Window or value? Attached with a stop-gradient to the weights of the headless policies, whose features the window loss never shaped, window heads add 3.4, 3.7, and 4.9 points on F2 to F4 to the conditioned OpenVLA-OFT, A2C2, and DR-$\pi_{0.5}$, within 0.6 points of the jointly trained heads (Table 6, Appendix D). A value head trained the same way on the same replays, which predicts the success of activating a chunk at a given delay and runs the option with the highest prediction, adds 3.3 and 4.7 points; window minus value lies within ±1.0 point on both backbones (90% intervals), inside the margin fixed in advance. The gain thus comes from the timing labels and a learned choice among chunks; the window is its compact form, two numbers per chunk that also say when to wait. *What the labels need* (development). A tuned constant window loses 5.0 points and a head without the timing-perturbed copies 1.9; half of the labeled demonstrations keep 4.4 of the head's 5.2 points, and event-time proxies or outcome labels 3.5 and 2.9. *Calibration and scheduling.* Upper-endpoint coverage is 53% on uncensored held-out block and copy labels and 80% on blocks alone, whose windows are wider; quantile heads reach 76% and 89% at nominal 75% and 90%. These are single-endpoint metrics (Appendix B). Replays at 2,000 logged ticks support ranking by window end. At 614 ticks after expiry, continuing and switching succeed in 57.8 and 58.5%; this comparison establishes neither equivalence nor constant continuation value (Appendix D).

6.4 When the dual-rate loops matter

In exploratory runs that add latency to the vision-language backbone alone (Table 14, Appendix D), the dual-rate policies lose 3.0 to 4.3 points from 0 to 400 ms, A2C2 9.3, and the conditioned OpenVLA-OFT 14.4, 10.1 [6.8, 13.4] more than DR-$\pi_{0.5}$; a dual-rate control whose action loop reads one fresh frame loses 3.5, so the robustness comes from keeping the backbone off the action

Table 6: What the labels buy. Left: confirmatory, three runs: head effect on F2 to F4 in points (95% intervals; window minus value, 90%, margin ± 2); plug-in heads use a stop-gradient on the weights retrained without the head. Right: development, three runs: success of DR-$\pi_{0.5}$ in % and difference from its joint window head (66.1%).

Head, effect on F2 to F4	OFT, conditioned	DR-$\pi_{0.5}$
Jointly trained window head	3.6 [2.2, 5.0]	5.5 [4.0, 7.0]
Plug-in window head	3.4 [2.3, 4.4]	4.9 [3.8, 5.9]
Plug-in value head	3.3 [2.3, 4.4]	4.7 [3.7, 5.8]
Window − value, 90%	0.1 [−0.8, 1.0]	0.1 [−0.8, 1.0]
Joint − plug-in window	0.3 [−1.2, 1.7]	0.6 [−0.8, 2.1]

Variant of DR-$\pi_{0.5}$	DYN	Δ [95% interval]
Retrained without head	60.9	−5.2 [−8.0, −2.3]
Tuned constant window	61.1	−5.0 [−7.8, −2.2]
Head without timing-perturbed copies	64.1	−1.9 [−4.3, 0.4]
Head on event-time proxies	64.4	−1.7 [−4.1, 0.7]
Head on outcome labels	63.8	−2.3 [−4.6, 0.1]
Oracle window	68.4	+2.3 [0.2, 4.4]
Oracle value	71.6	+5.5 [3.5, 7.5]

Table 7: Claims and evidence (confirmatory: three runs of 6,000 fresh episodes, plan fixed in advance; development: three runs of 1,200, exploratory; hardware: pre-registered paired batches). Points, 95% intervals.

Claim	Evidence
Latency limits a PhysClock-Syn-tuned $\pi_{0.5}$ (§4)	development, fixed weights: pausing the world adds 14.4, 15.5, 14.5
The head raises success on F2 to F4 (§6.2)	confirmatory: 3.6 [2.2, 5.0], 5.5 [4.0, 7.0]; all seven heads 3.4 to 5.5, $p < 0.001$
… mostly where timing is tight	confirmatory: head × TC 4.8 [2.3, 7.4], 6.7 [4.1, 9.3]; F1 zero within ± 2
Window and value heads are equivalent (§6.3)	confirmatory: window − value 0.1 [−0.8, 1.0] (90%), both backbones
The head helps on hardware (§6.5)	sheet 113 vs 79 of 280 (pooled post hoc), ball 60 vs 41 of 120; constant window 66 of 200

path, as in HiRT and πR^2. Without the 30 Hz loop, DR-$\pi_{0.5}$ loses 1.4 points [−0.9, 3.8] at about half the accelerator time.

6.5 Real robot

On an AgileX PiPER arm, all models are fine-tuned on the same 100 teleoperated demonstrations per task; in the pre-registered batches 3 to 5, both arms of each head contrast start from simulation weights trained without the window loss, and heads use event-time proxies from the demonstration videos (Appendix E, Table 17). In block-randomized batches with a blinded operator and annotators, the head raises DR-$\pi_{0.5}$'s sheet grasps from 79 to 113 of 280 paired trials over batches 3 and 4 (pooled post hoc, exact McNemar $p = 0.0003$; 58 to 83 of 200 in batch 4, Holm-adjusted $p = 0.011$ over the six primary contrasts) and its ball catches from 41 to 60 of 120 in batch 3 (Holm $p = 0.017$); on OpenVLA-OFT, from 53 to 72 of 200 sheets (Holm $p = 0.055$). A tuned constant window on the headless weights, a timing gate that learns nothing, grasps 66 of the 200 sheets (head against it, $p = 0.016$): it closes nearly as late as the head but inside the feasible interval less often (146 against 160 of 200), and less successfully when it does (45 against 52%). Pretraining on PhysClock-Syn rather than on the quasi-static counterparts raises ball catches from 45 to 63 of 120 and sheet grasps from 34 to 49 of 120 ($p = 0.013$ and 0.011).

7 Discussion and Conclusion

Fine-tuned on dynamic demonstrations and run synchronously, $\pi_{0.5}$ still loses 14 points to latency; timing labels, used to choose among chunks, recover part of it on two backbones, in simulation and on hardware. *Limits.* The labels assume one decisive action per episode with a known feasible interval, and windows help only near it. The best labels come from replays a real robot cannot run: a head trained in simulation does not transfer to the real arm (Appendix E), and on a held-out family the head changes success by 0.5 points [−1.9, 2.9], so each new task needs its own labels. The timing-critical subset comes from one simulator with a coarse air model, the ball result rests on one hardware batch, and DR-$\pi_{0.5}$ uses five times the accelerator time of synchronous $\pi_{0.5}$.

AI USE STATEMENT

In this work, we used generative AI tools only for language editing and polishing of the manuscript text, a task with recommended disclosure. We have not used generative AI tools for generating synthetic datasets, developing theoretical models, formulating mathematical claims, designing experiments, implementing methods, or interpreting results. We have reviewed all AI-assisted text for accuracy and fidelity to our results. 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 uses simulation and a controlled laboratory robot experiment with lightweight objects; no human subjects or personal data are involved. Faster-committing robot policies can cause harm through premature actions; our scheduler holds the current target whenever no valid action is available, and physical deployment used joint velocity limits and an operator-controlled stop.

REPRODUCIBILITY STATEMENT

We will release PhysClock, PhysClock-Syn with its tolerance and window labels, the labeling and evaluation code including the virtual clock, the latency profiles, the failure-mode classifier, and training configurations for all methods. The supplementary material contains an anonymized repository with the air solver, the virtual clock, and the labeling code, a labeled subset of PhysClock-Syn with a datasheet and licenses, the analysis plans with their hashes, per-episode outcomes for every table, rollout videos, and photographs of the real setup. Appendix A specifies the environment, the labels, and the metrics, Appendix B the window head, the scheduler, and the baselines, and Appendix D the confirmatory protocol.

REFERENCES

Ayoub Agouzoul. Understanding Asynchronous Inference Methods for Vision-Language-Action Models. *arXiv preprint arXiv:2605.08168*, 2026.

Anders Andersen, Umberto Pesavento, and Z. Jane Wang. Analysis of transitions between fluttering, tumbling and steady descent of falling cards. *Journal of Fluid Mechanics*, 541:91–104, 2005.

Akhil Bagaria, Ben Abbatematteo, Omer Gottesman, Matt Corsaro, Sreehari Rammohan, and George Konidaris. Effectively learning initiation sets in hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Mohak Bhardwaj, Thomas Lampe, Michael Neunert, Francesco Romano, Abbas Abdolmaleki, Arunkumar Byravan, Markus Wulfmeier, Martin Riedmiller, and Jonas Buchli. Real-World Fluid Directed Rigid Body Control via Deep Reinforcement Learning. In *Learning for Dynamics and Control Conference (L4DC)*, 2024. arXiv:2402.06102.

André Biedenkapp, Raghu Rajan, Frank Hutter, and Marius Lindauer. TempoRL: Learning when to act. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 914–924, 2021.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In *Conference on Robot Learning (CoRL)*, pp. 17–40, 2025a. arXiv:2504.16054.

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, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Robotics: Science and Systems (RSS)*, 2025b. arXiv:2410.24164.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-Time Execution of Action Chunking Flow Policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025c. arXiv:2506.07339.

Kevin Black, Allen Z. Ren, Michael Equi, and Sergey Levine. Training-Time Action Conditioning for Efficient Real-Time Chunking. *arXiv preprint arXiv:2512.05964*, 2025d.

Yann Bouteiller, Simon Ramstedt, Giovanni Beltrame, Christopher Pal, and Jonathan Binas. Reinforcement Learning with Random Delays. In *International Conference on Learning Representations (ICLR)*, 2021.

Chongkei Chang and Zhidong Deng. DynaWM: A Base-VLA-Guided World Foundation Model for Moving-Object Manipulation. *arXiv preprint arXiv:2607.02604*, 2026.

Baiming Chen, Mengdi Xu, Liang Li, and Ding Zhao. Delay-aware model-based reinforcement learning for continuous control. *Neurocomputing*, 450:119–128, 2021.

Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, and Pheng-Ann Heng. Fast-in-Slow: A Dual-System VLA Model Unifying Fast Manipulation within Slow Reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2506.01953.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, Honghao Su, Zhiyang Dou, Kaixuan Wang, Dandan Zhang, Yunze Liu, Yan Qin, Qiwei Liang, Qiwei Wu, Zijian Lin, Wenwei Lin, Yuran Wang, Minghua He, Tianshu Wu, Ruihai Wu, Jingquan Zhou, Kai-Chong Lei, Haibao Yu, Yuanfeng Ji, Weiyang Jin, Guanyu Lin, Xiaofan Li, Qi Xiong, Renjing Xu, Zhongyu Li, Wenhao Chai, Enze Xie, Ziwei Wang, Yao Mu, Hao Dong, Wojciech Matusik, Mingyu Ding, Wenbo Ding, Ping Luo, and Masayoshi Tomizuka. RoboDojo: A Unified Sim-and-Real Benchmark for Comprehensive Evaluation of Generalist Robot Manipulation Policies. *arXiv preprint arXiv:2607.04434*, 2026a.

Yuxuan Chen, Wanruo Zhang, and Xiao Li. Reflex: Enabling fast and predictive vision-language-action models for reaction-critical manipulation. *arXiv preprint arXiv:2608.14379*, 2026b.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Robotics: Science and Systems*, 2024.

Esther Derman, Gal Dalal, and Shie Mannor. Acting in delayed environments with non-stationary Markov policies. In *International Conference on Learning Representations*, 2021.

Perry Dong, Kuo-Han Hung, Dorsa Sadigh, and Chelsea Finn. Reinforcement learning for real-time vision-language-action policies. *arXiv preprint arXiv:2609.18207*, 2026.

Zhenyang Feng, Jimin Heo, Erik B. Sudderth, and Unnat Jain. TEMPO: Learning temporal context for dynamic robot manipulation. In *Conference on Robot Learning*, 2026.

Figure AI. Helix: A vision-language-action model for generalist humanoid control. `https://www.figure.ai/news/helix`, 2025.

Vlad Firoiu, Tina Ju, and Joshua Tenenbaum. At human speed: Deep reinforcement learning with action delay. *arXiv preprint arXiv:1810.07286*, 2018.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. ManiSkill2: A Unified Benchmark for Generalizable Manipulation Skills. In *International Conference on Learning Representations (ICLR)*, 2023.

E. Douglas Jensen, C. Douglass Locke, and Hideyuki Tokuda. A time-driven scheduling model for real-time operating systems. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pp. 112–122, 1985.

Armin Karamzade, Kyungmin Kim, Montek Kalsi, and Roy Fox. Reinforcement learning from delayed observations via world models. In *Reinforcement Learning Conference*, 2024.

Konstantinos V. Katsikopoulos and Sascha E. Engelbrecht. Markov Decision Processes with Delays and Asynchronous Cost Collection. *IEEE Transactions on Automatic Control*, 48(4):568–574, 2003.

Sanjit Kaul, Roy Yates, and Marco Gruteser. Real-time status: How often should one update? In *IEEE INFOCOM*, pp. 2731–2735, 2012.

Jangwon Kim, Hangyeol Kim, Jiwook Kang, Jongchan Baek, and Soohee Han. Belief projection-based reinforcement learning for environments with delayed feedback. In *Advances in Neural Information Processing Systems*, 2023.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. In *Conference on Robot Learning (CoRL)*, 2024. arXiv:2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Robotics: Science and Systems (RSS)*, 2025. arXiv:2502.19645.

Seungsu Kim, Ashwini Shukla, and Aude Billard. Catching objects in flight. *IEEE Transactions on Robotics*, 30(5):1049–1065, 2014.

Xunyao Lei, Renjun Wu, Tianlin Huo, and Xuesong Li. SparkVLA: Stop-aware hierarchical VLA with adaptive action chunking for long-horizon manipulation. *arXiv preprint arXiv:2608.16172*, 2026.

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*, 2026a.

Mengtian Li, Yu-Xiong Wang, and Deva Ramanan. Towards Streaming Perception. In *European Conference on Computer Vision (ECCV)*, pp. 473–488, 2020.

Yun Li, Jiachen Gong, Simon Thompson, Ehsan Javanmardi, Qunli Zhang, Zifan Zeng, Shiming Liu, Peng Wang, Zixuan Guo, and Manabu Tsukada. Think at 5 Hz, act at 20 Hz: Asynchronous fast-slow vision-language-action inference for closed-loop driving. *arXiv preprint arXiv:2607.15621*, 2026b.

Haoran Liao, Pengyue Wang, Shuoyu Chen, et al. DynamicManip: Enabling dynamic manipulation from a single static demonstration. *arXiv preprint arXiv:2608.01452*, 2026.

Haitao Lin, Yanwei Fu, and Xiangyang Xue. PourIt!: Weakly-supervised liquid perception from a single image for visual closed-loop robotic pouring. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 241–251, 2023.

Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. SoftGym: Benchmarking Deep Reinforcement Learning for Deformable Object Manipulation. In *Conference on Robot Learning (CoRL)*, 2020. arXiv:2011.07215.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. arXiv:2306.03310.

Mohan Liu, Dengchen Mei, Haotian Xian, et al. MotionForge: A data generation pipeline and large-scale benchmark for long-horizon manipulation of dynamic objects with domain shifts. *arXiv preprint arXiv:2609.25689*, 2026a.

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)*, 2025.

Zhe Liu, Jinghua Hou, Yuxiang Lu, Zhenya Yang, Xianzhe Fan, Junwei Luo, Junyi Li, Ruihua Han, Zhi Hou, and Hengshuang Zhao. StreamPI: Streaming Multimodal Temporal Modeling for Vision-Language-Action Models. *arXiv preprint arXiv:2608.26067*, 2026b.

Timothy A. Mann, Daniel J. Mankowitz, and Shie Mannor. Time-regularized interrupting options. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1350–1358, 2014.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. *arXiv preprint arXiv:2503.14734*, 2025.

Sungjae Park and Shubham Tulsiani. $\pi\mathbf{R}^2$: Reactive Real-time Flow Policies. *arXiv preprint arXiv:2607.26055*, 2026.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. In *Robotics: Science and Systems (RSS)*, 2025. arXiv:2501.09747.

Daniel E. Quevedo and Vijay Gupta. Sequence-based anytime control. *IEEE Transactions on Automatic Control*, 58(2):377–390, 2013.

Krithi Ramamritham. Real-time databases. *Distributed and Parallel Databases*, 1(2):199–226, 1993.

Simon Ramstedt and Chris Pal. Real-Time Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Matt Riemer, Gopeshh Raaj Subbaraj, Glen Berseth, and Irina Rish. Enabling Realtime Reinforcement Learning at Scale with Staggered Asynchronous Inference. In *International Conference on Learning Representations (ICLR)*, 2025.

Yaniv Romano, Evan Patterson, and Emmanuel Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Connor Schenck and Dieter Fox. Visual Closed-Loop Control for Pouring Liquids. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. arXiv:1610.02610.

Kohei Sendai, Maxime Alvarez, Tatsuya Matsushima, Yutaka Matsuo, and Yusuke Iwasawa. Leave No Observation Behind: Real-time Correction for VLA Action Chunks. *arXiv preprint arXiv:2509.23224*, 2025.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A Vision-Language-Action Model for Affordable and Efficient Robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Junhyuk So, Chiwoong Lee, Shinyoung Lee, Jungseul Ok, and Eunhyeok Park. Improving generative behavior cloning via self-guidance and adaptive chunking. In *Advances in Neural Information Processing Systems*, 2025.

Haoming Song, Delin Qu, Yuanqi Yao, Qizhi Chen, Qi Lv, Yiwen Tang, Modi Shi, Guanghui Ren, Maoqing Yao, Bin Zhao, Dong Wang, and Xuelong Li. Hume: Introducing system-2 thinking in visual-language-action model. *arXiv preprint arXiv:2505.21432*, 2025.

Yuteng Sun, Haoran Wang, Ruofei Bai, Zhengguo Li, Jun Li, Meng Yee Michael Chuah, and Wei Yun Yau. TIDAL: Temporally interleaved diffusion and action loop for high-frequency VLA control. *arXiv preprint arXiv:2601.14945*, 2026.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999.

Shahram Najam Syed, Arthur Jakobsson, Haoran Hao, and Jeffrey Ichnowski. Intercepting the Future: Latent-Space Predictive World Model for Dynamic VLA Manipulation. *arXiv preprint arXiv:2606.02486*, 2026.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5026–5033, 2012.

Jaden B. Travnik, Kory W. Mathewson, Richard S. Sutton, and Patrick M. Pilarski. Reactive reinforcement learning in asynchronous environments. *Frontiers in Robotics and AI*, 5:79, 2018.

David Valensi, Esther Derman, Shie Mannor, and Gal Dalal. Tree search-based policy optimization under stochastic execution delay. In *International Conference on Learning Representations*, 2024.

Thomas J. Walsh, Ali Nouri, Lihong Li, and Michael L. Littman. Learning and planning in environments with delayed feedback. *Autonomous Agents and Multi-Agent Systems*, 18(1):83–105, 2009.

Wei Wang, Dongqi Han, Xufang Luo, and Dongsheng Li. Addressing signal delay in deep reinforcement learning. In *International Conference on Learning Representations*, 2024a.

Yian Wang, Juntian Zheng, Zhehuan Chen, Zhou Xian, Gu Zhang, Chao Liu, and Chuang Gan. Thin-Shell Object Manipulations With Differentiable Physics Simulations. In *International Conference on Learning Representations (ICLR)*, 2024b. arXiv:2404.00451.

Yujin Wang, Junli Chen, Yixuan Li, Shunan Dong, Huazhong Yang, Yongpan Liu, and Hongyang Jia. The speedup paradox: Rethinking inference speed-quality trade-off in embodied tasks. *arXiv preprint arXiv:2606.28529*, 2026.

Yueyang Weng, Xiaopeng Zhang, Yongjin Mu, Yingcong Zhu, and Yanjie Li. Temporal action selection for action chunking. *arXiv preprint arXiv:2511.04421*, 2025.

Yansong Wu, Huaqing Li, Tianding Hou, Lingyun Chen, and Alois Knoll. React When You Need To: Event-Triggered Asynchronous Inference for VLA Policies. *arXiv preprint arXiv:2609.22587*, 2026.

Zhou Xian, Bo Zhu, Zhenjia Xu, Hsiao-Yu Tung, Antonio Torralba, Katerina Fragkiadaki, and Chuang Gan. FluidLab: A Differentiable Environment for Benchmarking Complex Fluid Manipulation. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2303.02346.

Ted Xiao, Eric Jang, Dmitry Kalashnikov, Sergey Levine, Julian Ibarz, Karol Hausman, and Alexander Herzog. Thinking While Moving: Deep Reinforcement Learning with Concurrent Control. In *International Conference on Learning Representations (ICLR)*, 2020.

Haozhe Xie, Beichen Wen, Jiarui Zheng, Zhaoxi Chen, Fangzhou Hong, Haiwen Diao, and Ziwei Liu. DynamicVLA: A Vision-Language-Action Model for Dynamic Object Manipulation. *arXiv preprint arXiv:2601.22153*, 2026.

Weichen Xu, Zhenhua Liu, Lin Luo, Yaobo Liang, Chengtang Yao, Qingyu Mei, Jian Cao, Xixin Cao, Xing Zhang, Jiaolong Yang, and Baining Guo. Continue or replan? Bernoulli-continuation policy learning for adaptive horizon execution. *arXiv preprint arXiv:2608.03483*, 2026.

Zhenjia Xu, Cheng Chi, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. DextAIRity: Deformable Manipulation Can be a Breeze. In *Robotics: Science and Systems (RSS)*, 2022. arXiv:2203.01197.

Shenhao Yan, Ge Wang, Qi Liu, Weilin Meng, Jiahao Yang, Chengsi Yao, Fan Feng, Xiaoguang Ma, Yiming Zhao, and Yatong Han. Acting While Understanding: Asynchronous Semantic-Action Decoupling for Real-Time Vision-Language-Action Models. *arXiv preprint arXiv:2606.15285*, 2026.

Yufeng Yuan and A. Rupam Mahmood. Asynchronous reinforcement learning for real-time control of physical robots. In *International Conference on Robotics and Automation (ICRA)*, 2022.

Haichao Zhang, Wei Xu, and Haonan Yu. Generative planning for temporally coordinated exploration in reinforcement learning. In *International Conference on Learning Representations*, 2022.

Jianke Zhang, Yanjiang Guo, Xiaoyu Chen, Yen-Jen Wang, Yucheng Hu, Chengming Shi, and Jianyu Chen. HiRT: Enhancing Robotic Control with Hierarchical Robot Transformers. In *Conference on Robot Learning (CoRL)*, 2024.

Yuanhang Zhang, Tianhai Liang, Zhenyang Chen, Yanjie Ze, and Huazhe Xu. Catch It! Learning to Catch in Flight with Mobile Dexterous Hands. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. arXiv:2409.10319.

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.

Xingyu Zhu, Wenshuo Han, Zhouyu Wang, Yuran Wang, Ruihai Wu, Hao Dong, Fan Tang, Hechang Chen, Hyung Jin Chang, and Yixing Gao. FlatLab: A Unified Methodology Framework and Simulation-Based Benchmark for Robotic Manipulation of Flat Objects. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2608.14049.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A Modular Simulation Framework and Benchmark for Robot Learning. *arXiv preprint arXiv:2009.12293*, 2020.

A Environment, Labels, and Metrics

Simulator. The environments use MuJoCo 3.2.3 with robosuite and LIBERO, a single Franka arm, and external and wrist cameras at 256×256 pixels, resized to 224×224 for the backbones. The control rate is 20 Hz and the camera rate 30 Hz; the physics step is 1 ms for the sheet and 2 ms otherwise. F1 pours 300 to 1,500 rigid spheres of 4 to 6 mm (up to 1,200 in training), whose contact damping and weak cohesion set how the stream spreads; F2 disturbs stacks of three to five blocks or cups by an impulse; F3 releases a ball onto a ramp and a slightly inclined table; F4 couples a thin shell to an air solver. Dynamic events start 0.5 to 2.0 s into an episode. The air solver is a staggered grid of 48^3 cells of 2 cm, advanced by semi-Lagrangian advection and a pressure projection and coupled two-way to the shell through a drag closure: each face receives $\frac{1}{2}\rho C_n A|v_n|v_n$ along its normal and $\frac{1}{2}\rho C_t A|v_t|v_t$ along the face, with the velocities taken relative to the air, plus the pressure difference across it; $C_n = 1.1$ and $C_t = 0.05$ were fitted to the fall times of 20 of 40 filmed real drops. The closure has no circulation lift or added mass (a variant with both is in Appendix D). Gusts are low-frequency velocity fields injected at one face, with amplitudes up to 1.0 m/s in training. The sheet is a $150 \times 200 \times 0.1$ mm PET film (4.1 g, bending stiffness 4×10^{-4} N m, Young's modulus 4.0 GPa, Poisson ratio 0.38) with a 1 g corner tab, the measured values of the real film, modeled as a `flexcomp` of 11×15 vertices; the 20×16 mm finger pads span one to two vertex spacings. The simulated gripper closes in 120 ms and the PiPER's in 180 ms, a 2 cm end-effector step has a 10–90% rise time of 60 ms in simulation and 85 ms on the PiPER, and blurring the rendered frames over a 10 ms exposure changes DR-$\pi_{0.5}$'s F4 success by -0.7 points.

Clock and regimes. The simulator keeps a virtual clock independent of wall time. A call issued at t reads the newest frame and returns at $t+\ell$; the returned chunk takes effect at the first control tick at or after $t+\ell$, from its time-aligned step, and an empty buffer holds the pose. By default ℓ is replayed from 1,000 calls of each policy, or of each loop of DR-$\pi_{0.5}$, recorded on one A800 at batch size one; sweeps inject a constant instead. QS replaces moving entities by their static terminal configurations and pauses during inference, PW keeps the dynamic scene but pauses during inference, and DYN advances physics throughout. The development evaluation replays profiles recorded with each component alone, charges each policy its own preprocessing and transport, and draws latency offsets per seed, a seed-level latency component; with offsets drawn per episode, every three-run contrast moves by at most 0.8 points. The confirmatory evaluation gives every policy a shared frame buffer, replays profiles recorded in each policy's deployment configuration, draws offsets per episode, and adds a 40 ms sensing and a 20 ms actuation delay, which lower development success by 2.0 to 3.8 points when added there.

Tasks, splits, and success. Each family has five dynamic task instances and five quasi-static counterparts. PhysClock-Syn holds 500 demonstrations per dynamic task, 10,000 in all, split 90/5/5 by episode, stratified by task; every training label comes from training demonstrations or from rollouts on training initial conditions, validation initial conditions serve tuning, and the development evaluation uses the initial conditions of the first 20 test demonstrations of each task, 400 in all, under three evaluation seeds that re-sample the stochastic physics, the latency offsets, and the policies' sampling noise. Success requires 70% of the particles in the cup with at most 10% spilled (F1), lifting and holding the target without toppling the rest (F2), grasping and lifting the ball (F3), or grasping the sheet with both pads before its lowest vertex touches the table (F4). The feasible interval opens when the stream leaves the spout (F1), the target enters a graspable pose (F2), the ball enters the reachable region (F3), or the lowest vertex of the sheet passes 30 cm above the table (F4), and closes at the deadline events of Table 2. The quasi-static counterparts of F1 and F4 are different tasks (placing a cup, grasping a flat sheet).

Expert and tolerance labels. A scripted expert per family reads ground truth, has no latency, and looks ahead by rolling out a copy of the simulator with the episode's random state; it triggers the decisive action when the look-ahead state first meets a family criterion, at a median of 0.42 of the feasible interval. The decisive segment runs from the expert's last replanning, 150 to 300 ms before the decisive action, to 200 ms after it. To label a demonstration, this segment is shifted by $\Delta \in \{-400, -350, \ldots, 400\}$ ms under the same seed with the earlier segments fixed, holding the pose in the gap that a positive shift opens, while the closed-loop expert plans the later segments from where the shifted one ends. The largest contiguous successful interval containing zero, refined at each end by two bisections to 12.5 ms, gives the tolerance. A window that spans the grid is recorded as 800 ms, and one that reaches an end of the grid is a lower bound, which holds for 98, 59, 26, and 8% of the tolerances in F1 to F4; relabeled on a $\pm$1,000 ms grid, the medians are 1,020, 700, 380, and 280 ms, since every lower bound exceeds 400 ms. Labeling runs MuJoCo with eight threads, and the 0.2% of replays at $\Delta = 0$ whose outcome differs from the demonstration are relabeled single-threaded; evaluations run single-threaded and are reproducible given their seeds and latency offsets. Of the 10,000 expert trials, 601 fail without a shift and are replaced under new seeds; compared directly with the kept F4 trials, the 259 failures in F4 fall faster (a median lowest-corner speed of 0.91 against 0.83 m/s at 30 cm) and tumble more often (39 against 27%), so the kept demonstrations under-represent fast tumbling drops. Each evaluation episode takes the tolerance of the expert run from its initial condition and seeds, or of the expert's next successful run where that fails (74 of the 1,200 development episodes); the timing-critical subset holds the 473 development episodes whose tolerance is below 400 ms (3 of them in F1), which no lower bound reaches.

Window labels and label sources. Chunks sampled every 100 ms from 1.0 s before to 0.9 s after the decisive action are replayed after holding for 0, 50, 100, 200, and 400 ms, activated time-aligned and continued by the expert, with two bisections between the last success of the first sampled successful interval and its next, larger failed delay (Algorithm 2); the smallest successful delay gives ϵ, and the last success before the next sampled failure gives $\delta_{\max}$, censored at 400 ms. When every delay fails, $\delta_{\max} = 0$ and ϵ is masked. Labels farther from the decisive action take the censoring value. Timing-perturbed copies move the decisive segment of a labeled chunk by $u \sim U(-300, 300)$ ms and use the same replays, plus two bisections between the first success and its

preceding failure to refine ϵ (zero if already successful at zero); they enter only the window loss, never the action loss, and make up half of its labels. A copy activated after its planned close issues the overdue gripper command at activation, so a premature copy can succeed when activated later; $\epsilon > 0$ for 2% of the block labels and 24% of the copy labels, with a median of 125 ms among positive labels. Labeling needs no policy inference and takes at most 161 replays per demonstration for the tolerance and the block windows, 1,479,640 in all (Table 8), and at most 90 more for its 10 timing-perturbed copies, 812,460 in all, about 3,400 CPU-core hours together. Event-time proxies assume the decisive step occurs at $\tau + d$. Under time alignment, a persistent close command is instead issued at $\max(t_o + d, \tau)$, ignoring control-tick discretization and actuation delay; skipped motion can also change the grasp pose. Thus these proxies approximate timing and do not certify task success. They need only t_f, t_d, and the decisive-step offset d; among the blocks and copies of the 500 test demonstrations, on the 6,140 uncensored labels of chunks that contain the decisive step, they differ from replay labels by a median of 31 ms for δ (Spearman 0.83). This agreement is an empirical property of these demonstrations, not a consequence of the definitions: for $s < d$ a time-aligned replay keeps the close at $t_o + d$, whereas the proxy moves it to $t_o + s + d$. Outcome labels come from 20,000 rollouts of DR-$\pi_{0.5}$ without the head: a chunk activated s after t_o whose decisive action landed on time, late, or early bounds $\epsilon \le s \le \delta$, $\delta < s$, or $\epsilon > s$ (censored Huber loss). Relabeling with β the previously active chunk moves $\delta_{\max}$ by a median of $+11$ ms, with DR-$\pi_{0.5}$ as continuation by -52 ms; across five re-sampled futures the window ends vary with a standard deviation of 33 ms.

Algorithm 2 Window labels for a chunk a computed from the frame at t_o (sketch; censoring at 400 ms)

Require: decisive step of a at $t_o + d$ (if any); feasible interval $[t_f, t_d]$ of the episode

1: **Replay labels:** hold for each $s \in \{0, 50, 100, 200, 400\}$ ms, activate a time-aligned, then continue with the expert; record success. If none succeeds, set $\delta_{\max} = 0$ and mask ϵ.
2: Otherwise find the first contiguous run of successful sampled delays; set ϵ to its first sample. Bisect twice between its last success and the next larger failure, keeping the successful side as $\delta_{\max}$; if it reaches 400 ms, right-censor there.
3: **Timing-perturbed copy:** apply the same replays; also refine ϵ by two bisections against the preceding failed delay, keeping the successful side (or $\epsilon = 0$ if zero succeeds).
4: **Event-time proxies (also for copies):** $\tilde{\epsilon} \leftarrow \max(0, t_f - t_o - d)$, $\tilde{\delta} \leftarrow t_d - t_o - d$; translated-chunk approximation, not exact replay bounds.
5: **Outcome labels (own rollouts):** for a chunk activated s after t_o: on time $\Rightarrow$ $\epsilon \le s \le \delta_{\max}$; late $\Rightarrow$ $\delta_{\max} < s$; early $\Rightarrow$ $\epsilon > s$ (censored Huber loss)
6: Chunks without a decisive step use event-time proxy targets $(0, 400)$ ms; replay labels outside $[-1.0, 0.9]$ s around the decisive action use $(\epsilon, \delta_{\max}) = (0, 400)$ ms (censored).

Re-grasps and the late side. The two confirmatory analyses of this paragraph were added after the plan. The expert's continuation may re-grasp after a close that misses, and a replay that succeeds this way counts as a success. Such rescues are rare: 1.2, 3.8, 5.9, and 2.1% of the successful shifted replays in F1 to F4 needed a second close, or in F1 a second pass under the stream. With re-grasps disabled, 93% of tolerances are unchanged and the mean shortens by 9 ms (F3 median 380 to 368 ms, F4 280 to 276 ms); the timing-critical subset grows from 473 to 496 development episodes and from 2,368 to 2,489 confirmatory episodes per run, and on it the confirmatory head effects are 5.8 [3.9, 7.7] points on the conditioned OpenVLA-OFT and 8.5 [6.7, 10.3] on DR-$\pi_{0.5}$, against 5.9 and 8.7. Latency delays the decisive action, so the late side of the tolerance interval matters most: its median is at least 400 ms in F1 (censored) and 330, 196, and 144 ms in F2 to F4, and on the 2,102 confirmatory episodes per run whose late side is below 200 ms the head adds 6.2 [4.2, 8.2] and 9.1 [7.1, 11.1] points.

Metrics. The on-time rate (OTR) is the fraction of episodes whose decisive action lands in $[t_f, t_d]$; with several attempts the last one counts, which concerns 2 to 4% of the episodes of each method. Lateness is the decisive time minus the deadline, observation age the time from the newest frame behind an executed step to its issue, and hold the fraction of control time without a new step. A failure-mode classifier assigns failed episodes to no attempt, early or late commit, wrong intercept (more than 3 cm off), or execution failure; it agrees with human annotation on 92 of 100 episodes. For replay labels $\delta_{\max,j}$ and predictions $\hat{\delta}_j$, *upper-endpoint coverage* is the fraction with $\delta_{\max,j} \ge \hat{\delta}_j$

Table 8: Demonstration and tolerance-labeling statistics. Expert success is a count over 2,500 trials per family; tolerance median and interquartile range (IQR) are in milliseconds; duration is the mean demonstration length in seconds; the next columns give the fraction of decisive actions with tolerance below 300 and 190 ms, the fraction of tolerances that are lower bounds, and the logged replays per family. Tolerances are censored at 800 ms, the width of the shift grid; $\geq$ marks medians of lower bounds.

Family	Expert success	Duration (s)	Median (ms)	IQR (ms)	<300 ms	<190 ms	Lower bounds	Replays
F1	2,433/2,500	6.0	$\geq$715	167	0.7%	0.0%	98%	335,040
F2	2,366/2,500	5.0	$\geq$626	369	9.0%	1.3%	59%	360,670
F3	2,359/2,500	4.0	380	208	27.7%	4.2%	26%	386,930
F4	2,241/2,500	3.5	280	173	56.1%	19.4%	8%	397,000

and MAE the mean absolute difference. This does not measure containment of the full predicted window, which additionally requires $\epsilon_j \leq \hat{\epsilon}_j \leq \hat{\delta}_j$. A censored label counts as covered when the prediction lies below it, so we also report both metrics on uncensored labels. Since bisection resolves a label only to r (12.5 to 50 ms), a label with $\delta_{\max} < \hat{\delta} < \delta_{\max} + r$ is a tie that may or may not be covered, and labels with $\delta_{\max} = 0$ are never covered. For quantile heads, a randomized probability integral transform (PIT) places each label within the piecewise-linear distribution through the five predicted quantiles (0.1, 0.25, 0.5, 0.75, 0.9), drawing uniformly within a tie; a calibrated head gives uniform values.

B Window Head, Scheduler, and Baselines

What the window head estimates. Let E and D be the ends ϵ and $\delta_{\max}$ of a chunk's realized window under the behavior of the labels (hold, execute the chunk, then the labeling policy), random through the future of the world; the labels are one draw of (E, D) per chunk, and the head sees only its input features x. Since the median of $g(D)$ is g of the median of D for a monotone g, the Huber loss on $\log(x + 50\,\mathrm{ms})$ with a threshold of 0.05, small against the spread of the transformed labels (a standard deviation of about 0.4), approximates median regression: the head estimates the medians of E and D given x over the chunks it is trained on, blocks and copies in equal parts, and a pinball loss at level q estimates the q-quantile. *Proposition.* If in every realization the successful activation times form the interval $[t_o + E,\, t_o + D]$, and $\hat{\epsilon} \geq Q_{1-\alpha/2}(E \mid x)$ and $\hat{\delta} \leq Q_{\alpha/2}(D \mid x)$ for the exact conditional quantiles, then every $\tau \in [t_o + \hat{\epsilon},\, t_o + \hat{\delta}]$ satisfies $\Pr(\tau \notin [t_o + E, t_o + D] \mid x) \leq \Pr(E > \hat{\epsilon} \mid x) + \Pr(D < \hat{\delta} \mid x) \leq \alpha$: the predicted interval lies inside the level-α window defined conditional on x. A learned head only approximates these quantiles; the interval assumption holds for 98.9% of labeled block starts. *Upper-endpoint calibration* (Table 9). The default head covers 53% of the uncensored held-out labels of blocks and copies (56% counting half of the ties), close to the 50% of a median, and 80% of the block labels alone, whose windows are wider than those of copies (uncensored medians of 176 against 118 ms): a head that cannot fully tell a block from a copy of similar timing predicts below the block's median. A constant per family shows the same pattern (50 and 71%), and one fitted on blocks alone covers 52% of them. Labels with $\delta_{\max} = 0$ (6.4% of block labels) are never covered; without them the head covers 85%. Without copies the head covers 57% of block labels but predicts $\hat{\epsilon} > 0$ for 0.4% of chunks instead of 11%. For the upper endpoint, quantile heads cover 76 and 89% of block and copy labels at nominal 75 and 90%; their randomized PIT is close to uniform on blocks and copies (Kolmogorov–Smirnov distance 0.04) and concentrated in the upper half on blocks (0.21, mean 0.66). A split-conformal correction of the (0.1, 0.9) head (Romano et al., 2019) reaches 90.3% upper-endpoint coverage on blocks and copies and changes success by -0.3 points $[-2.6, 2.0]$ on run 1: the scheduler relies on the order of the windows more than on their level.

Inputs and training. The head is a two-layer MLP of width 256. For a flow-matching expert it reads the hidden states of one extra expert pass at flow time 1 whose action input is the finished chunk, pooled over the action tokens (2 ms); OpenVLA-OFT's head reads the pooled states of its parallel decoding pass. The window loss has weight 0.1; labels censored at 400 ms are treated as exact, and a loss that penalizes a prediction only when it lies below a censored label changes success by 0.3 points. Trained jointly, the window loss reaches the expert through the extra pass, and the jointly trained expert executed without its head reaches 57.9% against 60.9% for the model trained

Table 9: Upper-endpoint accuracy against replay labels of the 500 held-out demonstrations; coverage does not test the full window. The upper rows count a censored label as covered when the prediction lies below it; the lower rows use uncensored labels, of blocks unless stated (ties and the PIT are defined in Appendix A). On the out-of-range physics split, MAE is 105 ms and coverage 78%.

Condition	MAE (ms)	δ coverage
All families; F4	80; 120	84%; 76%
Injected prefix latency of 200, 400, 800 ms	84, 92, 110	83, 81, 76%
Censored labels, F1 / F2 / F3 / F4		61% / 38% / 14% / 5%
Head	97	80%
ties / strictly below / labels with $\delta_{\max} = 0$		7% / 13% / 6.4%
blocks and copies (ties)	–	53% (6%)
Best constant per family; on blocks and copies; fitted on blocks	131	71%; 50%; 52%
Head trained without copies	94	57%
Head on event-time proxies; outcome labels; stop-gradient	112; 121; 104	77%; 74%; 78%
Quantile heads at nominal 75 / 90%, blocks and copies; blocks		76 / 89%; 91 / 97%
Split-conformal (0.1, 0.9) head, blocks and copies; blocks		90.3%; 97.6%
Within-family Spearman ρ, F1 to F4		0.41 / 0.58 / 0.63 / 0.55

without the loss, so we measure the head against the model retrained without it, or on shared weights with a stop-gradient head. Reading the expert at flow time 0, which hides the chunk from the head, costs 3.1 points on run 1.

DR-$\pi_{0.5}$ and compute. Fine-tuning uses the released `pi05_libero` checkpoint, 30,000 steps for every backbone, and the openpi LIBERO recipe. The prefix loop runs at 3 Hz; the action loop is requested at 20 Hz and reaches about 15 Hz with a 65 ms expert call; the motion loop encodes the last 8 frames of each camera at 128×128 into 16 tokens at 30 Hz with causal convolutions and a small temporal Transformer (about 30M parameters), trained through the expert and optionally by a prediction loss (weight 0.1) on privileged state 100 to 500 ms ahead. Stale-prefix training draws the prefix image $U(0, 300)$ ms earlier than the rest of the input. The candidate buffer holds 8 chunks, and only the newest valid candidate is compared with the active chunk; comparing every valid candidate changes success by -0.1 points. At batch size one, the prefix takes 85 ms, the expert and head 65 ms, and the encoder 6 ms, a nominal demand of 1.42 accelerator-seconds per second (s/s), against 0.28 for synchronous $\pi_{0.5}$ (190 ms per call, k=10) and 1.60 for OpenVLA-OFT re-planning at every tick. In the deployment configuration, with the prefix loop in its own process and the expert requested at 10 Hz, the three loops share one accelerator at 81% utilization and DR-$\pi_{0.5}$ reaches 64.1% against 65.2% with the isolated profiles.

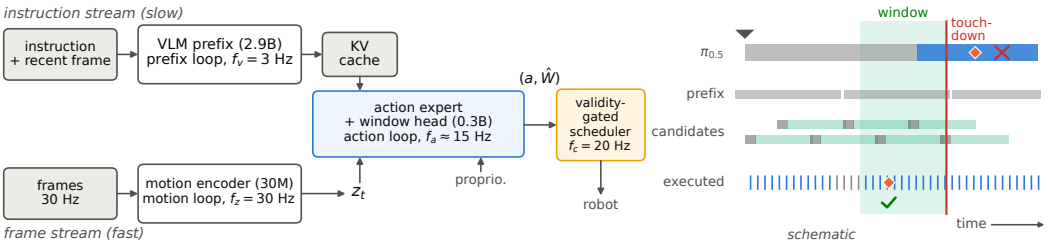

Figure 4: The dual-rate reference policy DR-$\pi_{0.5}$. Left: an instruction stream and a frame stream feed three asynchronous loops; the action expert and the window head return candidates $(a, \hat{W})$ to the scheduler of Algorithm 1. Right: schematic timeline; synchronous $\pi_{0.5}$'s chunk lands after touchdown, DR-$\pi_{0.5}$'s decisive step inside the window.

Baselines. All methods are fine-tuned on PhysClock-Syn for 30,000 steps from released checkpoints (`pi05_libero`, `pi0_fast_libero`, OpenVLA-7B with the OFT recipe, `smolvla_base`, and DynamicVLA's own) with the same demonstrations, control rate, and cameras. *RTC* (Black et al., 2025c) reuses the $\pi_{0.5}$ weights with the openpi implementation, soft-mask guidance with a maximum weight of 5, and the inference delay d of each call ($d = 4$ steps at 190 ms). Its configuration in Table 4 has execution horizon $s = d$ and latency augmentation (54.7% $\pm$ 0.8 over three runs, and 57.5% $\pm$ 0.8 with a window head whose inpainted candidates enter Algorithm 1, a head effect of 2.8 points [0.9, 4.8]); without augmentation it reaches 51.6%, and with $s = \max(10, d)$ 47.8% (run 1). *A2C2* (Sendai et al., 2025) adds a two-layer correction head

that reads the base action, the time index within the chunk, pooled SigLIP features of the latest frames, and optionally the 8-frame encoder features of DR-$\pi_{0.5}$, on $\pi_{0.5}$ with latency augmentation re-planning back-to-back; trained for 10,000 steps on the frozen base to predict the demonstrated action minus the time-aligned base step, it reaches 58.9%, and 61.0% with the 8-frame features (run 1). *Latency augmentation* delays the training observation by $U(0, 300)$ ms. *OpenVLA-OFT at every tick* issues an 80 ms call every 50 ms on two workers and executes the time-aligned steps of the newest chunk; the *conditioned* variant also receives its expected delay and the pending actions as tokens (Katsikopoulos & Engelbrecht, 2003; Bouteiller et al., 2021). *DynamicVLA's execution* on $\pi_{0.5}$ weights re-plans back-to-back and executes the time-aligned steps of the newest chunk. The *value head* shares the window head's architecture, also reads the activation delay, and predicts the success of activating the chunk after that delay, trained by binary cross-entropy on every replay of the window labels of blocks and copies (about 2.1 million outcomes); the scheduler runs the option with the highest value and holds if every value is below 0.5. The *keep-or-switch classifier* predicts which of two chunks 50 to 300 ms apart succeeds, and *quantile heads* replace the Huber loss by pinball losses. In development these heads read DR-$\pi_{0.5}$'s jointly trained features with a stop-gradient; the symmetric comparison trains window and value heads the same way on the weights retrained without the window loss. *Tuning.* Each control received 24 to 36 validation evaluations of 500 episodes, DR-$\pi_{0.5}$ 42, and the constant windows 77 ($\epsilon_c \in \{0, 50, \ldots, 300\}$, $\delta_c \in \{100, 150, \ldots, 600\}$ ms per family); a head adds nine to its arm, and nine more for each headless arm change its success by at most 0.2 points.

C Development Results

Degradation and diagnosis. Fine-tuned on the quasi-static counterparts, $\pi_{0.5}$, π_0-FAST, OpenVLA-OFT, and SmolVLA succeed in 862, 728, 750, and 650 of 1,200 QS episodes and in 178, 86, 288, and 264 DYN episodes. Pausing the world raises the QS-tuned $\pi_{0.5}$ from 14.8 to 24.4%, and halving its executed chunk as well to 39.1%. Fine-tuned on PhysClock-Syn, $\pi_{0.5}$ rises from 43.6 to 58.0% when the world pauses on run 1 and from 41.6 and 41.9 to 57.1 and 56.4% on runs 2 and 3, gains of 14.4, 15.5, and 14.5 points; pausing adds 8.2 points with latency augmentation (51.8 to 60.0%) and 2.2 for DR-$\pi_{0.5}$ with its head (65.2 to 67.3%). With the privileged state as extra tokens, $\pi_{0.5}$ reaches 68.9% in the paused world but only 47.8% in DYN. Table 10 gives the main development comparisons over three training runs.

Table 10: Development evaluation over three training runs (1,200 identical episodes per run): success in % per run, mean $\pm$ s.d., and DR-$\pi_{0.5}$ minus the row with its 95% interval over runs and episodes (Appendix D). Run 1 is the value of Table 4.

Method	Run 1	Run 2	Run 3	Mean $\pm$ s.d.	DR-$\pi_{0.5}$ $-$ row
DR-$\pi_{0.5}$ (window head)	65.2	66.9	66.2	66.1 ± 0.9	–
$\pi_{0.5}$, synchronous	43.6	41.6	41.9	42.4 ± 1.1	23.7 [19.6, 27.8]
OpenVLA-OFT, synchronous	55.5	58.6	56.3	56.8 ± 1.6	9.3 [5.6, 13.0]
$\pi_{0.5}$ + RTC ($s{=}d$, lat. aug.)	54.0	55.6	54.4	54.7 ± 0.8	11.4 [8.3, 14.6]
+ window head	56.9	58.4	57.2	57.5 ± 0.8	8.6 [5.4, 11.7]
$\pi_{0.5}$ + A2C2 (8-frame features)	61.0	62.1	62.9	62.0 ± 1.0	4.1 [0.4, 7.8]
OpenVLA-OFT, conditioned	61.8	63.4	63.2	62.8 ± 0.9	3.3 [0.1, 6.4]
+ window head	65.3	66.2	65.9	65.8 ± 0.5	0.3 [-2.9, 3.5]
DR-$\pi_{0.5}$ retrained without head	60.8	62.3	59.7	60.9 ± 1.3	5.2 [2.3, 8.0]
without motion encoder	58.8	59.1	61.8	59.9 ± 1.7	6.2 [2.7, 9.6]
executed synchronously	47.9	47.5	49.9	48.4 ± 1.3	17.6 [14.5, 20.7]
without stale-prefix training	57.3	57.2	56.2	56.9 ± 0.6	9.2 [6.4, 12.0]
without privileged prediction loss	65.1	66.7	65.8	65.9 ± 0.8	0.2 [-2.1, 2.6]
two loops, encoder in a 5 Hz action call	63.9	65.3	64.7	64.6 ± 0.7	1.4 [-0.9, 3.8]

Other backbones and generalization. Fine-tuned on PhysClock-Syn, π_0-FAST reaches 17.8% and SmolVLA with its asynchronous inference 35.3% in DYN; with the motion encoder, the head, and the scheduler, SmolVLA reaches 43.9%, 3.1 points of it from the head. On held-out physical parameters, appearance, and initial positions, DR-$\pi_{0.5}$ succeeds in 57.0, 62.7, and 60.9%, against 53.4, 61.5, and 59.1% for the conditioned OpenVLA-OFT with the head. With one family held out from fine-tuning in turn, DR-$\pi_{0.5}$ succeeds in 14.6%, and the head, whose windows cover only 52% of the uncensored block labels of the held-out family (80% in distribution), changes success by 0.5 points [-1.9, 2.9].

D CONFIRMATORY EVALUATION AND FURTHER ANALYSES

Statistics. Contrasts over three training runs are paired by episode and by run: every run is evaluated on the same episodes with the same latency offsets, and run i of every method uses the same training seed (i in development, $i + 3$ in the confirmatory evaluation). The variance of a three-run contrast adds the between-run variance of the paired per-run differences, divided by three, to the episode-level variance of the contrast on one run, from a bootstrap that resamples initial conditions with their seeds; the interval uses a t quantile with Satterthwaite degrees of freedom, capped at 1,000. Real-robot pairs use exact McNemar tests.

Confirmatory protocol. The 2,000 fresh initial conditions are 100 per task instance, drawn with a seed fixed before the evaluation and disjoint from the training, validation, and development conditions; each runs under three seeds with the protocol of Appendix A. The analysis plan, committed with its hash before any model was trained, fixed the primary endpoint, success on F2 to F4; the two primary contrasts, the head's effect on the conditioned OpenVLA-OFT and on DR-$\pi_{0.5}$, each against the same policy retrained without the head, Holm-corrected; a secondary Holm family of the four head effects, adding the delay-augmented OpenVLA-OFT and A2C2 with 8-frame features; three definitions of timing-critical episodes (the look-ahead expert's tolerance, a causal expert's tolerance, and the length of the feasible interval, which needs no replay), each with the 400 ms threshold; F1 as a negative control, tested by two one-sided tests against a margin of 2 points; and the comparisons across systems, with unadjusted intervals. An amendment, committed before any plug-in head was trained, added window and value heads trained with a stop-gradient for 5,000 steps on the weights of the conditioned OpenVLA-OFT, DR-$\pi_{0.5}$, and A2C2 retrained without the head (a window head on all three, a value head on the first two); the plug-in window effects form a Holm family of three, and window minus value on each backbone is tested against a margin of 2 points. All models were trained anew with training seeds 4 to 6 after the plan was fixed; no run was discarded and no episode excluded. The timing-critical subset holds 2,368, 2,561, and 2,214 episodes per run under the three definitions; excluding the 371 episodes whose expert run failed changes every timing-critical effect by at most 0.3 points.

Confirmatory results. On F2 to F4 the head adds 3.6 [2.2, 5.0] points to the conditioned OpenVLA-OFT (133 degrees of freedom), 5.5 [4.0, 7.0] to DR-$\pi_{0.5}$ (46), 4.0 [2.6, 5.4] to the delay-augmented OpenVLA-OFT (108), and 3.9 [2.5, 5.3] to A2C2 (370); Holm-adjusted p is below 0.001 for all four. By family, it adds 1.6, 4.3, and 4.9 points on F2, F3, and F4 to the conditioned OpenVLA-OFT and 2.8, 6.2, and 7.6 to DR-$\pi_{0.5}$. On timing-critical episodes the four effects are 5.9 [4.0, 7.8], 8.7 [6.9, 10.5], 6.3 [4.4, 8.1], and 5.9 [4.0, 7.8]; with the causal expert's tolerance the primary effects there are 5.7 [3.8, 7.6] and 8.4 [6.6, 10.2], and with the feasible-interval length 6.1 [4.1, 8.1] and 8.9 [7.0, 10.8]. On the other episodes of F2 to F4 they are 1.1 to 2.0 points; the head $\times$ TC interaction is 4.8 [2.3, 7.4] and 6.7 [4.1, 9.3] points for the primary contrasts. On F1 the 90% intervals of the four effects lie within ± 1.9, inside the equivalence margin. Across systems, DR-$\pi_{0.5}$ minus the conditioned OpenVLA-OFT with the head is -0.7 [-2.6, 1.2] on F2 to F4 (90% interval [-2.3, 0.9], so equivalence within 2 points is not established), and DR-$\pi_{0.5}$ minus single-loop $\pi_{0.5}$ with 8-frame tokens and the head is 3.3 [1.5, 5.1] (15 degrees of freedom). *Plug-in heads (amendment).* On the same weights, the plug-in window head adds 3.4 [2.3, 4.4] points on F2 to F4 to the conditioned OpenVLA-OFT, 3.7 [2.6, 4.8] to A2C2, and 4.9 [3.8, 5.9] to DR-$\pi_{0.5}$ (Holm-adjusted $p < 10^{-9}$), and nothing on F1 (90% intervals within ± 1.3); the jointly trained heads exceed them by 0.3 [-1.2, 1.7] and 0.6 [-0.8, 2.1] points. The plug-in value head adds 3.3 [2.3, 4.4] and 4.7 [3.7, 5.8] points, and window minus value is 0.1 point on both backbones with 90% intervals of [-0.8, 1.0], inside the margin. *Paired counts.* Table 12 gives the per-run discordant pairs of the primary contrasts. A conditional logistic regression added after the plan, which stratified by initial condition across runs and therefore pooled episodes of different models in each stratum, gave odds ratios near the marginal ones (1.31 against a marginal 1.28 on timing-critical episodes); stratified by episode within run, as the paired intervals are, it gives the matched-pair odds ratios of Table 12.

Uses of the labels (development). Table 13 evaluates rules on the weights of the three DR-$\pi_{0.5}$ runs and retrained variants. A constant window executes the newest valid chunk and differs from ignoring the head only in which candidates are valid; tuned per family, it loses 5.0 points. Of the head's two outputs, $\hat{\delta}$ carries most of the 8.2 points that the head adds on the same weights: setting

Table 11: Confirmatory evaluation per training run (training seeds 4 to 6): success in % on F2 to F4 (4,500 episodes per run), on timing-critical episodes (2,368), and on F1 (1,500), and the overall mean over runs (6,000 per run). Plug-in heads are trained with a stop-gradient on the weights of the row without the head of the same policy.

	F2 to F4, runs 1 / 2 / 3			Timing-critical, runs 1 / 2 / 3			F1, runs 1 / 2 / 3			Overall
DR-$\pi_{0.5}$	52.4	52.6	53.7	40.9	41.4	42.4	83.2	83.0	85.3	60.6
retrained without head	47.5	46.5	48.1	32.6	32.4	33.6	84.1	82.7	85.3	56.5
OpenVLA-OFT, tick, delay aug. + cond. + head	53.2	52.7	54.9	40.8	40.1	42.7	79.7	80.5	80.9	60.3
OpenVLA-OFT, tick, delay aug. + cond.	49.1	49.4	51.4	34.2	34.7	37.0	79.5	81.6	80.7	57.6
OpenVLA-OFT, tick, delay aug. + head	53.0	53.0	53.3	40.8	41.2	40.6	80.7	79.6	81.4	60.0
OpenVLA-OFT, tick, delay aug.	49.1	48.6	49.7	34.9	34.3	34.7	79.9	79.3	82.3	57.0
A2C2 + 8-frame features + head	52.6	52.2	52.7	41.5	40.6	41.5	81.4	81.5	82.2	59.8
A2C2 + 8-frame features	49.0	48.0	48.8	36.1	34.1	35.6	81.0	81.2	83.0	56.9
OpenVLA-OFT, conditioned, plug-in window head	52.5	52.8	54.8	39.4	40.2	42.1	79.7	81.7	80.5	60.2
OpenVLA-OFT, conditioned, plug-in value head	52.4	52.7	54.8	39.4	40.2	42.4	79.5	81.4	80.8	60.1
A2C2 + 8-frame features, plug-in window head	52.9	51.5	52.6	42.4	39.6	41.6	81.4	80.5	83.3	59.7
DR-$\pi_{0.5}$ without head, plug-in window head	52.3	51.6	52.8	40.3	40.8	41.3	83.7	82.9	85.5	60.2
DR-$\pi_{0.5}$ without head, plug-in value head	52.2	51.2	52.9	40.5	40.0	41.2	84.1	82.8	85.5	60.1

Table 12: Discordant pairs of the primary contrasts per training run, episodes paired within a run: with the head only / without the head only. The timing-critical subset holds 2,368 episodes per run (15 of them in F1), F2 to F4 4,500, and the other episodes of F2 to F4 2,147; about a fifth of the pairs are discordant.

Contrast	Run 1	Run 2	Run 3	Matched-pair OR
Conditioned OpenVLA-OFT, F2 to F4	510 / 326	501 / 355	517 / 359	1.47 [1.36, 1.59]
timing-critical	316 / 161	300 / 172	310 / 174	1.83 [1.64, 2.04]
other episodes of F2 to F4	196 / 167	203 / 185	209 / 187	1.13 [1.00, 1.27]
DR-$\pi_{0.5}$, F2 to F4	557 / 338	581 / 309	575 / 323	1.77 [1.63, 1.91]
timing-critical	350 / 155	359 / 146	351 / 143	2.39 [2.14, 2.67]
other episodes of F2 to F4	208 / 186	225 / 164	225 / 182	1.24 [1.10, 1.39]

$\hat{\epsilon}$=0 costs 2.8 of them and triples early commits. Selection alone (the slack rule, never discarding) keeps 4.1 of the 5.2 points over the retrained model, gating alone 1.9. In the last 200 ms before the decisive action a newer window ends a median of only 6 ms earlier than the active one while the head's error is about 100 ms, but consecutive errors are correlated (0.81), and on pairs whose replayed $\delta_{\max}$ differs by more than the label resolution $\hat{\delta}$ orders 74% correctly; rules that freeze switching near the decisive action without an ordering recover at most 1.2 of the 5.2 points. A value head, a keep-or-switch classifier, and quantile heads lie within 1.1 points of Algorithm 1. Without timing-perturbed copies the head loses 1.9 points and early commits rise from 13 to 38; with 50, 25, and 10% of the labeled demonstrations it loses 0.8, 1.9, and 3.3 points.

Table 13: Rules and label uses in the development evaluation (three runs): success of DR-$\pi_{0.5}$ in %, difference from Algorithm 1 with 95% interval, early commits (of 1,200, run 1), and chunk switches per second in the last 200 ms before the decisive action. Upper block: rules on the same weights; lower block: retrained or relabeled variants.

Rule or variant	DYN	Δ [95% interval]	Early	Switches/s
Algorithm 1: predicted window, slack rule	66.1 $\pm$ 0.9	–	13	5.8
Newest valid chunk, head ignored	57.9 $\pm$ 1.2	−8.2 [−10.5, −6.0]	11	15.1
Tuned constant window	61.1 $\pm$ 1.3	−5.0 [−7.8, −2.2]	9	14.0
Predicted $\hat{\delta}$, $\hat{\epsilon}$=0	63.3 $\pm$ 1.2	−2.8 [−4.8, −0.7]	41	6.0
Gating only (newest valid)	62.8 $\pm$ 0.8	−3.3 [−5.3, −1.2]	12	13.9
Selection only (no discard)	65.0 $\pm$ 0.5	−1.1 [−3.1, 1.0]	14	5.6
Slack of the active chunk + $(t - \tau^\star)$	65.8 $\pm$ 0.8	−0.3 [−2.3, 1.7]	13	4.2
Value head, highest predicted success	65.4 $\pm$ 0.8	−0.7 [−2.9, 1.5]	15	6.6
Keep-or-switch classifier	65.0 $\pm$ 0.5	−1.1 [−3.3, 1.1]	14	5.1
Quantile heads (0.25, 0.75); (0.1, 0.9)	66.0; 65.1	−0.1; −1.0	18; 26	5.4; 4.9
Oracle window (replayed labels of own candidates)	68.4 $\pm$ 0.5	+2.3 [0.2, 4.4]	11	4.9
Oracle value (replayed outcomes of every option)	71.6 $\pm$ 0.5	+5.5 [3.5, 7.5]	6	4.3
Retrained without head	60.9 $\pm$ 1.3	−5.2 [−8.0, −2.3]	12	15.0
Window head, stop-gradient on the weights without the window loss	65.7 $\pm$ 0.7	−0.4 [−2.8, 2.1]	13	6.0
Value head, stop-gradient on the same weights	65.2 $\pm$ 0.6	−0.9 [−3.3, 1.5]	15	6.7
Head without timing-perturbed copies	64.1 $\pm$ 0.9	−1.9 [−4.3, 0.4]	38	6.0
Head on 50 / 25 / 10% of the labeled demonstrations	65.3 / 64.1 / 62.8	−0.8 / −1.9 / −3.3	14 / 15 / 16	6.0 / 6.8 / 8.9
Head on event-time proxies	64.4 $\pm$ 0.8	−1.7 [−4.1, 0.7]	14	6.3
Head on outcome labels	63.8 $\pm$ 0.6	−2.3 [−4.6, 0.1]	16	6.8

Empirical checks of the scheduler. From the logged states of DR-$\pi_{0.5}$ on seed 1 of run 1, we sampled 2,000 ticks within 1 s of the decisive action, from 388 episodes, at which the active chunk and a newer valid candidate both existed, and replayed both options under the episode's random state. Both succeed in 1,212 cases, both fail in 468, only the switch in 171, and only the continuation in 149. In a logistic regression of the difference between the two outcomes on the difference of their predicted window ends, the switch becomes the better option at a difference of 5 ms [−21, 31], with a slope of 0.61 per 100 ms [0.31, 0.91], supporting a threshold near zero in Algorithm 1 on these sampled states; where exactly one option succeeds, the slack rule picks it in 204 of 320 cases (64%, 57 to 71%). At 614 further ticks at which the active chunk's predicted window had ended while it still had steps, continuing it succeeded in 57.8% of the replays and switching to the newest valid candidate in 58.5% (−0.7 [−4.4, 3.0]). This interval establishes neither equivalence nor preservation of the value at activation: the comparison is between two options after expiry, not across the life of an active chunk. Intervals resample episodes with all their ticks.

Added backbone latency. For Table 14, a constant is added to every prefix call of the dual-rate policies and to every call of the others, whose single forward pass contains the backbone; multi-worker baselines add workers to keep their call rate, and delay augmentation or stale-prefix training is extended by the added latency. OpenVLA-OFT decodes its actions through every layer of its language model, so the dual-rate controls are built on $\pi_{0.5}$: one whose action loop reads one fresh frame of each camera through a one-frame version of the encoder, closest to HiRT and πR^2, and one that reads only the prefix image and so has no fresh visual path. Every cell is three training runs of 1,200 episodes with latency offsets drawn per episode. Without heads, the loops lose 3.5 points from 0 to 400 ms, the conditioned OpenVLA-OFT 14.3, and $\pi_{0.5}$ re-planning back-to-back 16.8. Under a uniform slowdown of every component, where the action path slows with the backbone, DR-$\pi_{0.5}$ and the conditioned OpenVLA-OFT with its head differ by -0.2 to 0.5 points across slowdowns of 0.5 to 3 (intervals of about $\pm$3.5 points).

Table 14: Success in % (three runs, offsets drawn per episode) with latency added to the backbone as described in the text. Drop from 0 to 400 ms with 95% interval; the drop of the conditioned OpenVLA-OFT exceeds that of DR-$\pi_{0.5}$ by 10.1 points [6.8, 13.4], treating the two drops as independent. All rows carry the head.

Added backbone latency	0 ms	100 ms	200 ms	400 ms	Drop
DR-$\pi_{0.5}$	66.4	66.0	64.6	62.1	4.3 [1.9, 6.6]
Two loops, encoder in a 5 Hz action call	64.7	64.4	63.2	60.5	4.2 [1.8, 6.5]
Dual-rate, one fresh frame in the action loop	63.2	62.9	62.0	59.7	3.5 [1.1, 5.8]
Dual-rate, prefix image only (no fresh visual path)	60.3	60.1	59.2	57.3	3.0 [0.6, 5.3]
A2C2 + 8-frame features	65.2	62.9	60.3	55.9	9.3 [6.8, 11.9]
OpenVLA-OFT, conditioned	66.0	63.2	59.0	51.6	14.4 [12.0, 16.7]
DR-$\pi_{0.5}$ − OpenVLA-OFT, conditioned	0.4 [−3.1, 3.9]	2.8 [−0.7, 6.4]	5.6 [2.0, 9.1]	10.5 [7.0, 14.0]	
DR-$\pi_{0.5}$ − one fresh frame	3.2 [0.8, 5.6]	3.1 [0.7, 5.6]	2.6 [0.2, 5.0]	2.4 [0.0, 4.8]	

Simulated and real sheets. We dropped the real sheet 40 times from 0.6 to 0.8 m, filmed at 120 fps, and simulated 400 drops with matched release heights and orientations, with gusts off as the laboratory air is still (Table 15); comparisons use Welch t tests on all 40 real drops (the 26 fluttering ones for the flutter period). The drag coefficients were fitted to the fall times of 20 real drops; on the other 20 the fall time is 0.83 $\pm$ 0.08 s against 0.82 $\pm$ 0.07 s (p=0.59). The interval that sets the timing of F4, from the lowest corner passing 30 cm above the table to touchdown, lasts 352 $\pm$ 74 ms on the real sheet and 366 $\pm$ 69 ms in simulation (p=0.26). Over the last 300 ms before touchdown, the default closure matches the descent speed of the lowest corner (p=0.21) but moves sideways more slowly (p=0.03), pitches less (p=0.01), and flutters more slowly (0.31 against 0.27 s, $p < 0.001$); the variant with a quasi-steady circulation lift and added mass (Andersen et al., 2005) matches all four quantities (p from 0.34 to 0.75), and under it the head's F4 effect holds (below). A 5.1 g sheet falling broadside would reach about 1.6 m/s; real and simulated sheets tilt by 30 to 50° and glide, so the normal drag also resists the descent and the lowest corner passes grasp height at 0.8 to 0.9 m/s. Real pours of 5 mm glass beads, the middle of F1's range of 4 to 6 mm, reach cup height after 0.30 $\pm$ 0.04 s against 0.28 $\pm$ 0.03 s in simulation (p=0.04); 30 real ball rolls decelerate at 0.34 $\pm$ 0.05 against 0.31 $\pm$ 0.04 m/s^2 in 200 simulated ones (p=0.003), so simulated balls roll slightly farther, and 20 disturbed real stacks settle after 2.1 $\pm$ 0.4 against 1.9 $\pm$ 0.3 s (p=0.04).

Table 15: Real and simulated sheet drops from 0.6 to 0.8 m (400 simulated drops each, gusts off, with the 1 g corner tab), with the default drag closure and with a quasi-steady lift and added mass, and real and simulated pours of 5 mm beads (20 and 200); mean $\pm$ standard deviation.

Quantity	Real, 40 drops	Simulated	Lift and added mass
Fall time (s)	0.84 ± 0.08	0.82 ± 0.07	–
Lateral drift at touchdown (cm)	11.2 ± 6.3	9.4 ± 5.1	–
Fluttering drops (lateral oscillation above 3 cm)	65%	61%	66%
Lowest corner from 30 cm to touchdown (ms)	352 ± 74	366 ± 69	–
Tilt 100 ms before touchdown, median (°)	38	34	–
Lateral speed of the centroid, last 300 ms (m/s)	0.29 ± 0.11	0.25 ± 0.10	0.28 ± 0.11
Peak-to-peak pitch, last 300 ms (°)	48 ± 16	41 ± 15	46 ± 16
Flutter period, fluttering drops (s)	0.27 ± 0.05	0.31 ± 0.06	0.28 ± 0.05
Lowest-corner descent speed at 30 cm (m/s)	0.87 ± 0.19	0.83 ± 0.17	0.86 ± 0.18
Pour: stream reaches cup height after tilt onset (s)	0.30 ± 0.04	0.28 ± 0.03	–
Pour: stream width at cup height (cm)	2.8 ± 0.5	2.6 ± 0.4	–
Pour: beads spilled (%)	5.9	5.2	–

Robustness of F4 to the air model. F4 holds 240 of the 473 timing-critical episodes, so the timing results depend on the sheet model. Table 16 re-labels and re-evaluates F4 (run 1, 300 episodes per method) under six variants: the drag closure in still air without the grid, the closure coefficients scaled by 1.25 and by 0.75, gusts switched off, a quasi-steady lift with added mass, and a film collision radius halved to 0.2 mm. The tolerance median moves between 266 and 318 ms, the Jaccard index between the timing-critical F4 episodes and the default ones is 0.83 to 0.94, and the head's F4 effect stays between 7.3 and 8.7 points on DR-$\pi_{0.5}$ and between 3.0 and 4.0 on the conditioned OpenVLA-OFT; with 300 episodes each effect has a standard error of about 3 points, so the variants do not separate from one another. Retrained under the lift and added-mass model, which matches the real sheet's kinematics best (Table 15), the effects are 7.9 [3.1, 12.7] and 3.9 [-0.8, 8.6] points over three runs. With heads on both, DR-$\pi_{0.5}$ leads the conditioned OpenVLA-OFT on F4 by 2.3 to 3.7 points under every variant.

Table 16: F4 under variants of the air model: tolerance median and IQR (ms), Jaccard index of the timing-critical F4 episodes with the default, and F4 success in % with and without the window head (retrained without it; run 1, 300 episodes each), with the head effects in points.

Air model	Tolerance	Jaccard	DR-$\pi_{0.5}$, head / none	OFT cond., head / none	Effect, DR-$\pi_{0.5}$	Effect, OFT
Default (grid, closure, gusts)	280 (173)	–	37.3 / 29.3	34.0 / 30.3	8.0	3.7
Closure only, no grid	292 (168)	0.89	36.0 / 28.0	33.3 / 29.7	8.0	3.7
Closure coefficients $\times 1.25$	266 (176)	0.91	38.3 / 29.7	34.7 / 30.7	8.7	4.0
Closure coefficients $\times 0.75$	297 (171)	0.88	35.3 / 27.7	32.3 / 29.0	7.7	3.3
No gusts	318 (121)	0.83	40.3 / 33.0	38.0 / 34.3	7.3	3.7
Quasi-steady lift and added mass	289 (170)	0.90	36.7 / 29.0	33.3 / 30.0	7.7	3.3
Film collision radius 0.2 mm	276 (172)	0.94	35.7 / 28.3	32.3 / 29.3	7.3	3.0

Experts. The look-ahead expert succeeds in 93.8% of the development episodes; reading only the current state it reaches 87.5%, with DR-$\pi_{0.5}$'s timing 79.7%, and with $\pi_{0.5}$'s latency and chunking 54.3%; re-sampling its future costs 1.8 points (4.3 in F4). Fine-tuned on demonstrations from the re-sampling expert, whose chunks carry no knowledge of the realized future, DR-$\pi_{0.5}$ gains 5.1 points [2.7, 7.5] from the head and the conditioned OpenVLA-OFT 2.8 [0.4, 5.1], so the head's effect does not rest on future information in the demonstrations.

E Real-Robot Details

Platform and tasks. Both tasks use an AgileX PiPER arm (6-DoF, 626 mm reach) with its standard parallel gripper, joint velocities limited to 1.5 rad/s, and an operator stop. Two Intel RealSense D435i cameras, one in front of the workspace and one on the wrist, record 640×480 images at 30 fps, and a 120 fps side camera films the drops and the demonstrations; the policy commands end-effector pose deltas at 20 Hz. The sheet is the 150×200 mm PET film of Appendix A with a 1 g steel tab, hung flat by the tab from an electromagnet, long side horizontal, at one of five marked yaw angles, and released from 0.6 to 0.8 m (the height of its lowest edge); sheets were replaced

after at most 110 drops or when creased. The ball, 50 mm and 45 g, is released by a servo latch from one of five positions on a 0.4 m ramp and enters the reachable region at 0.82 to 1.21 m/s, the two fastest positions above the simulated training range. A script starts the policy and triggers the release after a random delay of 1.5 to 2.5 s; inference runs on one RTX 4090. The PiPER is slower than the simulated Franka (about 0.9 against 1.7 m/s at the end effector); real success is lower than in simulation on the ball (50% against 65% in F3) but not on the sheet (42% against 37% in F4), whose real releases lack gusts.

Data and training. All models share 100 real demonstrations per task, collected by leader–follower teleoperation with a second PiPER, which succeeded in 100 of 163 sheet and 100 of 121 ball attempts, and are fine-tuned on them for 10,000 steps. In batches 3 to 5, every arm except the "real only" arm starts from simulation weights trained without the window loss (the PhysClock-Syn-tuned weights unless marked otherwise), so the arms of a head contrast differ only in the window loss applied during real fine-tuning, on event-time proxies derived from the 120 fps demonstration videos (the ball entering and leaving the reachable region; the lowest sheet corner passing 30 cm and touching the table). The constant-window arm of batch 4 runs the headless weights behind the window $[t_o + 100,\ t_o + 250]$ ms, the best of the grid of Appendix B on 40 real validation drops. The "real only" arm fine-tunes DR-$\pi_{0.5}$ from the released $\pi_{0.5}$ weights, and the quasi-static arm from weights fine-tuned on the quasi-static counterparts. Every arm is a single fine-tuning run; in simulation, fine-tuning DR-$\pi_{0.5}$ with and without the head on 100 F4 demonstrations under three fine-tuning seeds gives head effects of 11.0, 13.3, and 9.7 points (s.d. 1.9), small against the 12.5 points of batch 4. The head trained on event-time proxies covers 84% of ball exits and 81% of sheet touchdowns in batch 3. The frozen simulation head of batches 1 and 2 (40 paired trials per method, task, and batch) has only 58% upper-endpoint coverage on real trials; its predicted upper endpoints are too late, and its lower endpoints are a median of 64 ms too early. These batches support DR-$\pi_{0.5}$ over synchronous $\pi_{0.5}$ (31 against 15 balls of 80, p=0.0015) but cannot isolate the head.

Protocol. Each batch, with the trial counts of Table 17, followed a written plan whose SHA-256 hash was time-stamped before its first trial and which fixed the sample sizes, the primary contrasts, and their Holm family. The primary contrasts are the head's effects on DR-$\pi_{0.5}$ and its comparison with OpenVLA-OFT at every tick on both tasks in batch 3, the head's effects on DR-$\pi_{0.5}$ and OpenVLA-OFT on the sheet in batch 4 (sized for 86% power at $\alpha = 0.05$ for a 12-point difference with 30% discordant pairs), and the pretraining contrast in batch 5; the constant window, the pretraining comparison of batch 4, and the other comparisons across policies are secondary. Methods run in an order randomized within each release condition; the operator who arms the releases sees only a trial number, and two annotators score the videos without method labels, agreeing before resolution on 777 of 800 trials in batch 3, 1,221 of 1,240 in batch 4, and 236 of 240 in batch 5. Outcomes are also assigned from color-marker tracks alone, which agree with the adjudicated labels in 984 of the 1,000 sheet trials of batch 4. No trial was excluded, repeated, or re-scored; one operator stop in batch 3 counts as a failure.

Paired analysis. Each trial is paired with the trial of the other method from the same release condition; counts are given as both succeed, only the first, only the second, neither. *Batch 3.* On the ball, DR-$\pi_{0.5}$ with the head against itself without it gives 31, 29, 10, 50 (exact McNemar p=0.0034), against OpenVLA-OFT 32, 28, 10, 50 (p=0.0051), and against the real-only arm 29, 31, 8, 52 (p=0.0003); on the sheet, the first two give 13, 17, 8, 42 (p=0.11) and 12, 18, 8, 42 (p=0.076), with 36 to 42% power at the observed effect, which is why batch 4 was run. *Batch 4.* On the sheet, the head on DR-$\pi_{0.5}$ gives 40, 43, 18, 99 (p=0.0019) and on OpenVLA-OFT 33, 39, 20, 108 (p=0.018); Holm-corrected over the six primary contrasts of batches 3 and 4, p is 0.011 and 0.017 for the head on DR-$\pi_{0.5}$ (sheet, ball), 0.020 for DR-$\pi_{0.5}$ against OpenVLA-OFT on the ball, 0.055 for the head on OpenVLA-OFT, and 0.15 for the two sheet contrasts of batch 3. The head gives 52, 31, 14, 103 against the constant window (p=0.016), and the constant window gives 45, 21, 13, 121 against no gate (p=0.23). Measured from the video, the gripper closes a median of 47 ms (IQR 9 to 105) after t_f without a gate, 63 ms (24 to 118) with the constant window, and 76 ms (38 to 131) with the head; the close falls inside $[t_f, t_d]$ in 133, 146, and 160 of 200 trials, and those closes succeed in 44, 45, and 52%. The gate thus closes nearly as late as the head, but the head chooses chunks that close inside the interval more often and grasp better when they do. On the ball, PhysClock-Syn pretraining gives 30, 33, 15, 42 against quasi-static pretraining (p=0.013). *Pooled and tracker-only.*

Table 17: Real-robot outcomes of the pre-registered batches: success, early and late commit, wrong intercept, and execution failure, classified as in Appendix A, and OTR in %; "head" uses event-time proxies, and starting weights are given under Data and training.

Batch, task, arm	Success	Early	Late	Wrong	Execution	OTR
Batch 3, rolling ball, 120 trials						
DR-$\pi_{0.5}$ without head	41	11	29	25	14	67
OpenVLA-OFT, every tick	42	6	31	21	20	69
DR-$\pi_{0.5}$, real only, head	37	9	32	27	15	66
DR-$\pi_{0.5}$, head	60	7	17	17	19	80
Batch 3, falling sheet, 80 trials						
DR-$\pi_{0.5}$ without head	21	14	12	16	17	68
OpenVLA-OFT, every tick	20	9	19	18	14	65
DR-$\pi_{0.5}$, real only, head	16	8	21	13	22	64
DR-$\pi_{0.5}$, head	30	5	14	15	16	76
Batch 4, falling sheet, 200 trials						
DR-$\pi_{0.5}$ without head	58	31	36	41	34	67
DR-$\pi_{0.5}$ without head, tuned constant window	66	17	37	36	44	73
OpenVLA-OFT, every tick	53	29	46	29	43	63
OpenVLA-OFT, every tick, head	72	12	41	39	36	74
DR-$\pi_{0.5}$, head	83	9	31	33	44	80
Batch 4, rolling ball, 120 trials, head						
DR-$\pi_{0.5}$, quasi-static pretraining	45	8	31	23	13	68
DR-$\pi_{0.5}$, PhysClock-Syn pretraining	63	6	18	15	18	80
Batch 5, falling sheet, 120 trials, head						
DR-$\pi_{0.5}$, quasi-static pretraining	34	9	35	25	17	63
DR-$\pi_{0.5}$, PhysClock-Syn pretraining	49	7	24	18	22	74

In an analysis added after both plans, over the sheet trials of batches 3 and 4 stratified by batch, the head on DR-$\pi_{0.5}$ gives 53, 60, 26, 141 (113 against 79 of 280; p=0.0003). From the marker tracks alone, the primary head contrasts give 40, 42, 18, 100 on the sheet of batch 4 (p=0.0027) and 31, 28, 10, 51 on the ball of batch 3 (p=0.0051). *Batch 5.* PhysClock-Syn pretraining gives 26, 23, 8, 63 against quasi-static pretraining on the sheet (49 against 34 of 120; p=0.011).

F Positioning

Here the augmented state of a delayed MDP (Katsikopoulos & Engelbrecht, 2003; Bouteiller et al., 2021) holds the world, the active chunk with its due step, the returned candidates, and the calls in flight; the actions are to continue, to activate a candidate, or to hold, and the value of activating a candidate is $V_{\pi,\beta}$ of Eq. 1, with W^{α} its superlevel set. The window head compresses this value into two numbers per candidate, and the scheduler uses window end as a heuristic candidate score, as options (Sutton et al., 1999; Mann et al., 2014) are interrupted when an alternative has a higher value; a validity window is a temporal analogue of a learned initiation set (Bagaria et al., 2023), and delayed-execution MDPs that actuate the newest available decision (Valensi et al., 2024) formalize the rule the scheduler replaces. Reinforcement learning on the augmented state is practical for VLAs in simulation (Li et al., 2026a) but not on a robot that cannot replay, hence our labels from replays, event times, and rollouts. Robot catching plans interceptions from predicted flight (Kim et al., 2014; Zhang et al., 2025); our tasks share its deadlines, but the policy is a general VLA.