PLANNING BEFORE AND AFTER FEEDBACK: ALLOCATING FIVE COMPLETE PLANNER CALLS WITH FROZEN VISUAL WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Planning with a frozen visual world model poses a temporal allocation problem: given $K = 5$ complete planner calls, how many calls r should be spent on independent search before acting, and how many should be reserved to replan from fresh observations? We compare precommitted schedules while fixing the model, scorer, call count, and logical function evaluations within each panel. On Wall, a disjoint pilot selects $r = 3$; relative to the all-initial-search schedule $r = 5$, the registered endpoint changes by -34.7 (two-way 95% sensitivity interval $[-45.0, -24.5]$), with a descriptive success change of $+31.6$ percentage points. PointMaze provides a boundary: both registered intervals for selected $r = 1$ versus $r = 5$ include zero. A matched Wall sham replaces fresh post-execution observations with cached imagined latents while holding later call slots and horizon work fixed; real minus sham is -23.9 (two-way 95% sensitivity interval $[-31.9, -16.4]$), supporting fresh-observation conditioning relative to that control. The Wall direction recurs in targeted iCEM-component, DecentCEM-style, and one-checkpoint JEPA-WM adaptations. PushT repeats the fixed $r = 4$ versus $r = 5$ direction in a prospective train-split policy confirmation and a retrospective validation-split replication, although equal-call restart controls do not establish a task-general feedback mechanism. Planner-call timing is therefore a measurable, task-dependent controller choice under the tested protocols, not a universal replanning rule.

1 INTRODUCTION

A visual world-model controller evaluates candidate action sequences under predicted outcomes, executes an action or short prefix, and may optimize again after a new observation (Finn & Levine, 2017; Hafner et al., 2019; Zhou et al., 2025). This repeated interface leaves a basic design choice unresolved: with a fixed number of complete planner calls, should the controller spend them on independent search before acting, or reserve some calls for optimization after execution reveals new information? Closed-loop benchmarks increasingly expose inference-time scaling as an explicit experimental factor (Zhang et al., 2026a), yet call timing is often entangled with model changes, online triggers, or variable compute.

Adjacent work triggers replanning from plan likelihood or model mismatch (Zhou et al., 2023; Cheng et al., 2027), adapts action execution from prediction–observation consistency (Wang et al., 2026a), or reduces token computation within a rollout (Chun et al., 2026). We ask a narrower controlled question. The world model, goal encoder, scorer, optimizer, number of complete calls, and logical function evaluations remain fixed within a comparison; only a precommitted schedule determines when calls observe the system. Because that schedule also changes commitment and predicted horizon work, we estimate its total protocol-level effect and reserve mechanism language for a matched information control.

We use the cross-entropy method (CEM) as the sampling optimizer (de Boer et al., 2005). Our protocol fixes $K = 5$ complete CEM calls, and one macro-action spans five primitive steps. A schedule index $r \in \{1, \ldots, 5\}$ spends r independently seeded full-horizon calls at o_0, selects the lowest predicted-cost plan, and executes its first r macro-actions. Each remaining call replans once from a fresh observation over the remaining horizon before executing one macro-action. Thus $r = 5$

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.

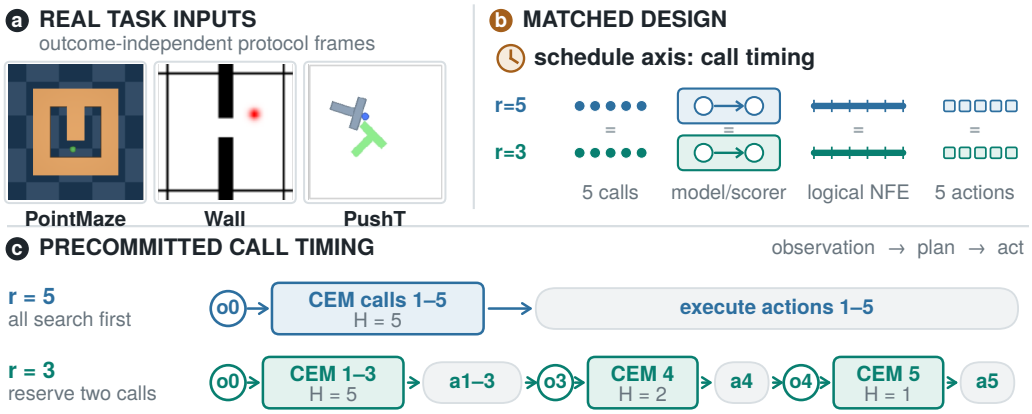


Figure 1: **Real visual inputs and precommitted planner-call timing.** Panel (a) shows outcome-independent real inputs from the frozen PointMaze, Wall, and PushT protocols; they are neither model outputs nor selected successes. Panel (b) visualizes the matched calls, model/scorer, logical NFE, and actions; call timing is the schedule axis. Panel (c) contrasts all-initial $r = 5$ against $r = 3$, which reserves calls 4 and 5 for fresh observations o_3 and o_4. Horizon-weighted work and synchronized planning time are not matched.

spends every call before acting, while smaller r reserves calls for later observations. Call count and logical NFE match, but horizon-weighted work and elapsed time need not. Figure 1 places the two timing regimes beside outcome-independent input examples from all three tasks.

We evaluate the full $r = 1, \ldots, 5$ curve on Wall and PointMaze and preserve pilot-selected contrasts. A matched Wall sham keeps later slots and horizon work fixed but replaces fresh observations with cached imagined latents. PushT supplies a prospective train-split allocation-policy confirmation and a retrospective validation-split replication. Three Wall adaptations test directional recurrence with iCEM components (Pinneri et al., 2021), a DecentCEM-style ensemble (Zhang et al., 2022), and one official JEPA-WM checkpoint (Terver et al., 2026); none is presented as an official end-to-end reproduction.

Contributions.

- **A controlled fixed-budget schedule study.** We formalize temporal placement of complete planner calls while keeping model, scorer, call count, and logical NFE fixed, and separately account for predicted-horizon work, synchronized time, and memory.
- **Task-dependent empirical evidence.** The pilot-selected Wall schedule lowers registered endpoint cost by 34.7, with a descriptive success increase of 31.6 percentage points relative to $r = 5$; both PointMaze intervals include zero. PushT repeats the fixed $r = 4$ versus $r = 5$ direction on two explicitly qualified panels.
- **Mechanism and boundary tests.** A matched Wall sham supports fresh-observation conditioning relative to cached imagined latents, while equal-call PushT controls and adaptive-policy evaluations prevent task-general feedback or adaptive-superiority claims.

2 Related Work

Visual world models for closed-loop planning. Action-conditioned video prediction and latent dynamics established visual model-predictive control from pixels (Finn & Levine, 2017; Hafner et al., 2019). Dreamer learns behavior through compact latent imagination, TD-MPC combines task-oriented latent dynamics with short-horizon optimization and terminal value prediction, and Masked World Models separates visual representation learning from latent dynamics for robot control (Hafner et al., 2020; Hansen et al., 2022; Seo et al., 2023). VLMPC instead pairs action-conditioned video prediction with a vision-language cost for robotic manipulation (Zhao et al., 2024). Recent systems

broaden this substrate: DINO-WM predicts pretrained patch features, V-JEPA 2-AC post-trains a video representation with robot interactions, and PLDM studies reconstruction-free latent objectives and MPC replanning intervals (Zhou et al., 2025; Assran et al., 2025; Sobal et al., 2025). JEPA-WM analyzes architecture, objective, data, and planning optimization; LeWorldModel develops a stable end-to-end JEPA for CEM-MPC; and World-in-World evaluates heterogeneous generative world models through closed-loop task success and inference-time scaling (Terver et al., 2026; Maes et al., 2026; Zhang et al., 2026a). HD-VPD provides a complementary physical representation based on RGB-D particle dynamics and candidate motion-plan evaluation (Whitney et al., 2025). We instead freeze the model and scorer and intervene on the precommitted timing of complete planner calls.

When to replan and how long to execute. Event-triggered MPC updates the optimizer when state–prediction error reaches a prescribed trigger (Li & Shi, 2014). Fixed-chunk policies expose a related but distinct execution choice: Diffusion Policy executes a fixed prefix before replanning, whereas ACT temporally ensembles overlapping multi-step chunks from frequent policy queries (Chi et al., 2023; Zhao et al., 2023). Diffusion replanning uses plan likelihood, while AdaReP adapts a tolerance from cached-rollout deviation and local sensitivity (Zhou et al., 2023; Cheng et al., 2027). World action models vary execution from future–reality consistency or monitored task progress (Wang et al., 2026a; Ye et al., 2026). VLA systems verify a proposed chunk, detect visual-latent drift, or learn continue-or-replan decisions under a frozen base policy (Wang et al., 2026c; Pan et al., 2026; Xu et al., 2026); AAC changes chunk length from action entropy, while RTC overlaps execution with generation of the next chunk (Liang et al., 2026; Black et al., 2025). These methods adapt execution or heavy-model calls online; our schedules fix both the number and placement of complete CEM calls before evaluation.

Other inference-allocation axes. AdaJEPA updates the world model between planning rounds (Wang et al., 2026b); Sparse Imagination varies visual-token computation within a rollout (Chun et al., 2026); and hierarchical latent planning changes temporal abstraction and macro-action structure (Zhang et al., 2026b). Our comparison instead matches complete calls and logical function evaluations, then measures horizon-weighted work, synchronized time, and memory rather than assuming equal physical compute.

Planner families and ranking reliability. CEM is a standard sampling optimizer for learned-dynamics MPC (de Boer et al., 2005; Bharadhwaj et al., 2020; Chua et al., 2018). iCEM uses temporally correlated sampling and memory, and DecentCEM maintains independently updated workers (Pinneri et al., 2021; Zhang et al., 2022). Proposal design is a separate axis: learned sampling distributions, behavior/previous-plan priors, and policy-network proposals change candidate quality at a fixed nominal sample budget (Sacks & Boots, 2023; Argenson & Dulac-Arnold, 2021; Wang & Ba, 2020). Our iCEM-component and DecentCEM-style tests are within-codebase adaptations, not official end-to-end reproductions; they exclude path-integral control, differentiable world-model planning, and diffusion trajectory planning (Williams et al., 2017; Psenka et al., 2026; Janner et al., 2022). More fundamentally, value equivalence distinguishes decision-relevant model accuracy from transition fidelity (Grimm et al., 2020). Closed-loop success therefore need not imply correct ranking of a shared candidate set: ARC-Bench separates those quantities; TRM changes terminal ranking with a fixed world model; and Traj-LeWM combines endpoint distance with a path-aware trajectory cost while also shaping its representation during training (Zhang & Li, 2026; Li et al., 2026; Huang et al., 2026). Our sham neither audits nor repairs action ranking; it tests only the information supplied to later calls.

3 Problem Setup

We study a pre-specified schedule family in a frozen goal-conditioned controller; we do not introduce a new world model, scorer, or online trigger.

Frozen controller and exact latent score. For task τ, let o_t be the current observation, g the goal observation, and a_t one macro-action (five primitive environment steps in the primary protocols). The released preprocessor normalizes task-native inputs, and the frozen encoder produces visual and proprioceptive goal latents, $z_g = E_\phi(g) = (z_g^{\mathrm{v}}, z_g^{\mathrm{p}})$. Given a candidate sequence $\mathbf{a}_{t:t+h-1}$, the

frozen predictor rolls forward from the permitted current observation and returns terminal latents $\hat{z}_{t+h}$. Candidates are ranked by

$$c_{\tau,\mathcal{M}}(\mathbf{a}; o_t, g) = D_{\mathrm{v}}(\hat{z}^{\mathrm{v}}_{t+h}, z^{\mathrm{v}}_g) + \alpha_{\tau,\mathcal{M}} D_{\mathrm{p}}(\hat{z}^{\mathrm{p}}_{t+h}, z^{\mathrm{p}}_g), \tag{1}$$

where, for $q \in \{\mathrm{v}, \mathrm{p}\}$,

$$D_q(\hat{z}, z) = \frac{1}{|\mathcal{I}_q|} \sum_{i \in \mathcal{I}_q} (\hat{z}_i - z_i)^2. \tag{2}$$

The index set $\mathcal{I}_q$ contains every non-batch terminal-latent coordinate. The frozen coefficients are $\alpha_{\textsc{PointMaze},\mathrm{DINO}} = 0$, $\alpha_{\textsc{Wall},\mathrm{DINO}} = \alpha_{\textsc{PushT},\mathrm{DINO}} = 1$, and $\alpha_{\textsc{Wall},\mathrm{JEPA}} = 0.1$. Thus PointMaze uses visual MSE, DINO-WM Wall/PushT sums visual and proprioceptive MSE, and JEPA-WM Wall uses visual plus 0.1 proprioceptive MSE. Physical terminal distance, success, and the registered endpoint below are unavailable to planning and selection; all scorers remain frozen.

One complete call. Let $\mathcal{P}(o_t, g; \xi, h) = (\mathbf{a}, c)$ denote one complete CEM invocation with seed ξ and remaining macro-horizon h. Standard CEM calls draw $N = 300$ trajectories per update, retain 30 elites, and score the returned final elite mean once. PointMaze/Wall use $I = 10$ updates, so their logical function-evaluation count is

$$Q = NI + 1 = 3{,}001. \tag{3}$$

PushT uses $I = 30$ with the same N and elite count, so $Q = 9{,}001$. The iCEM adaptation instead fixes 15,000 new scores per five-call arm and makes no post-selection scorer call. Every primary episode has $H = K = 5$ macro-actions/calls and always executes all five actions; success cannot stop search or execution early.

Registered task endpoint and analysis unit. After execution, the official evaluator returns task distance d_τ and success $s_\tau \in \{0, 1\}$. We report both and use the frozen scalar endpoint

$$J_\tau = d_\tau + \lambda_\tau (1 - s_\tau), \qquad \lambda_\tau = \begin{cases} 100, & \tau \in \{\textsc{PointMaze}, \textsc{Wall}\}, \\ 50, & \tau = \textsc{PushT}. \end{cases} \tag{4}$$

Lower is better. For task/model stack $\mathcal{M}$, schedule s_r indexed by r, scene or episode e, and registered planner-seed map m, let $J^{\tau,\mathcal{M}}_{e,m}(s_r)$ denote this endpoint. We average the three maps within each e before forming paired contrasts and uncertainty intervals. Scenes are the primary units for PointMaze/Wall and episodes for PushT; the crossed map cells are not independent replicates.

With the model, scorer, task, K, and per-call optimizer fixed, the remaining intervention is the temporal placement of calls.

4 Planning Allocation

The fixed budget creates a choice between search from the initial observation and optimization after execution has revealed a new observation. We estimate the total effect of this schedule bundle; no schedule is selected online.

Precommitted schedule family. The primary protocols use $H = K = 5$. For $r \in \{1, \ldots, H\}$, the first r calls share o_0, use distinct seeds, and optimize the full horizon:

$$(\mathbf{a}^{(j)}, c^{(j)}) = \mathcal{P}(o_0, g; \xi^{\mathrm{init}}_j, H), \qquad j = 1, \ldots, r. \tag{5}$$

The controller selects $j^* = \arg\min_{j \le r} c^{(j)}$ and executes the first r macro-actions of $\mathbf{a}^{(j^*)}$. For each later macro-time $t = r, \ldots, H-1$, it obtains the fresh permitted observation o_t, plans over the exact remaining horizon, and executes only the first macro-action (we use one-based indexing inside action sequences):

$$(\mathbf{a}_t, c_t) = \mathcal{P}(o_t, g; \xi^{\mathrm{fb}}_t, H - t), \qquad a_t = \mathbf{a}_t[1]. \tag{6}$$

The call-horizon sequence is

$$\mathbf{h}(r) = \left(\underbrace{H, \ldots, H}_{r \text{ initial calls}}, H - r, H - r - 1, \ldots, 1 \right), \tag{7}$$

with an empty suffix when $r = H$. Hence $r = 1$ replans after every macro-action, whereas $r = 5$ chooses among five initial plans and executes the selected plan without further observations.

Matched logical budget, measured physical work. For the standard-CEM panels governed by Eq. equation 3, every schedule uses K complete calls and

$$\mathrm{NFE}_{\mathrm{logical}}(r) = \sum_{h \in \mathbf{h}(r)} Q = KQ. \tag{8}$$

Later calls are shorter, so predicted trajectory work is

$$W(r) = Q \sum_{h \in \mathbf{h}(r)} h = Q \left[rH + \frac{(H-r)(H-r+1)}{2} \right]. \tag{9}$$

For PointMaze/Wall, $W(r)$ ranges from $45{,}015$ to $75{,}025$ despite constant logical NFE $15{,}005$. The locked PushT comparison has $W(4) = 189{,}021$ and $W(5) = 225{,}025$ at logical NFE $45{,}005$. We therefore report W, CUDA-synchronized planning time, batched rollout invocations, and peak allocated/reserved memory separately; "equal logical NFE" never means equal physical compute.

Coupling without arm-level credit. Within each scene–map pair, schedules share initialization and goal. Schedule r consumes the first r records of one frozen five-call initial sequence. Materializing those records once is exact because observation, goal embedding, checkpoint, and seeds are identical. This reduces collection duplication but does not change arm-level accounting: every arm is charged five logical calls, its own KQ, W, and measured planning time. Environment transitions are never shared after two schedules choose or execute different actions.

Changing r jointly changes initial search breadth, commitment length, observation access, later horizons, and W. The full curve is therefore a controlled schedule comparison, not a pure feedback-effect estimate. The matched Wall sham narrows the information contrast by holding the $r = 3$ execution and resource contract fixed while replacing fresh later observations with cached imagined latents.

5 Experimental Protocol

Tasks, models, and information boundary. We evaluate released task-specific DINO-WM checkpoints on PointMaze, Wall, and PushT (Zhou et al., 2025), plus one released JEPA-WM Wall checkpoint (Terver et al., 2026). Checkpoints, encoders, goal representations, scorers (Eqs. 1–2), and optimizer hyperparameters remain frozen. Planning receives only permitted RGB, evaluator-supplied proprioception, the goal, and candidate actions. Simulator state and future outcomes never enter scoring, selection, or early stopping; success is evaluated only after all five macro-actions. Appendix A gives the full panel ledger.

Selection and evidence roles. PointMaze and Wall each have a 32-scene pilot and a disjoint 96-scene full set, crossed with three pre-specified planner-seed maps. The pilot selects r by mean J_τ over scenes and maps, breaking ties toward smaller r; the full set cannot revise the choice. This yields $r = 1$ for PointMaze and $r = 3$ for Wall. The primary full-set estimand is selected r minus $r = 5$; the complete five-point curve is also reported. A 24-reset PushT development curve fixes $r = 4$. A previously unused 64-episode train-split bank is then consumed once for prospective allocation-policy confirmation against $r = 5$. Because those episodes may have contributed to world-model training, they are not a representation holdout. The same locked contrast is evaluated on all 21 validation episodes; prior diagnostic access makes this panel retrospective and non-blind. No panel permits outcome-dependent removal, extension, or retuning.

Mechanism and robustness controls. The Wall sham fixes $r = 3$ and matches scene, goal, model/scorer, initial calls, selected plan, executed prefix, later slots/horizons, seeds, CEM settings, logical NFE, W, and actions. It substitutes cached imagined latents for fresh post-execution RGB/proprioception; later optimizer trajectories may diverge. Two post-hoc PushT controls compare feedback with full-horizon restarts at three and five calls. The shared Wall panel also tests iCEM-component (Pinneri et al., 2021), three-worker DecentCEM-style (Zhang et al., 2022), and one-checkpoint JEPA-WM adaptations. They are not independent task replications or official end-to-end reproductions; iCEM uses separate 15,000-new-score accounting.

Inference and resource accounting. We average the three maps within each scene or episode before forming paired contrasts. Primary 95% intervals use scene- or episode-block percentile bootstraps; crossed scene/episode–map bootstraps are separate three-map sensitivities. Paired sign-flip calculations use the same scene/episode units and are model-based sensitivity analyses, not treatment-randomization tests. Success changes are secondary and descriptive unless accompanied by an interval or test. We separately report logical NFE, W, batched rollout invocations, CUDA-synchronized planning time, and peak allocated/reserved memory; shared calls remain fully charged to every logical arm. Appendix A gives draw counts, corrections, multiplicity handling, and the exact-enumeration case.

6 Results

Overview. The effect of temporal planner-call placement is task dependent under these protocols. With model, scorer, call count, and logical NFE fixed within each comparison, PointMaze does not establish an advantage over $r = 5$, whereas Wall and both PushT panels favor schedules that reserve calls for later observations. Figure 2 shows every PointMaze/Wall schedule as a contrast against $r = 5$ and the two locked PushT contrasts. Primary intervals resample scenes or episodes after averaging the three maps; crossed intervals are secondary three-map sensitivities.

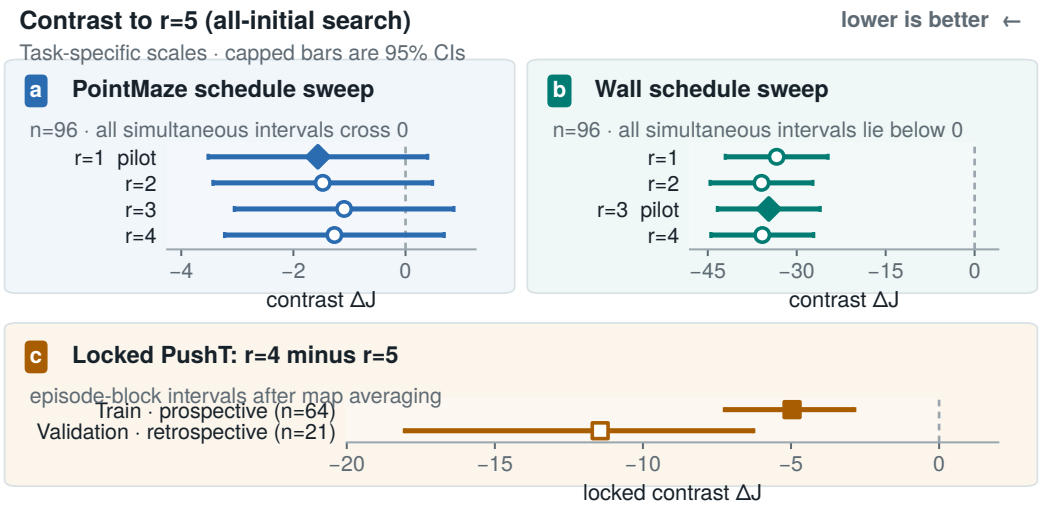


Figure 2: **Temporal placement has task-dependent effects.** PointMaze and Wall rows show $J_r - J_5$ with simultaneous 95% scene-block intervals after averaging three pre-specified maps; diamonds mark the disjoint-pilot choices. PushT shows the locked $r = 4 - r = 5$ contrasts with primary 95% episode-block intervals after map averaging. The train-split panel confirms the allocation policy rather than held-out representations; the validation panel is retrospective and non-blind. All arm pairs match $K = 5$ and logical NFE, not horizon-weighted work or time. Lower is better.

6.1 PointMaze bounds the effect; Wall favors reserving calls

The PointMaze pilot selected $r = 1$. On the 96-scene full panel, $r = 1 - r = 5$ is -1.56 (primary 95% scene-block interval $[-3.59, 0.09]$; two-way sensitivity interval $[-4.33, 0.73]$), with a descriptive success change of $+1.4$ percentage points and Holm-adjusted sign-flip sensitivity $p = 0.0619$. All four simultaneous $r < 5$ contrasts include zero. The panel does not establish an improvement over $r = 5$, but neither does it establish equivalence.

The Wall pilot selected $r = 3$. On the disjoint full panel, $r = 3 - r = 5$ is -34.71 (primary 95% interval $[-42.78, -26.66]$; two-way sensitivity interval $[-45.03, -24.53]$), with the same direction in all three maps. Success increases by a descriptive 31.6 percentage points and the Holm-adjusted sensitivity is $p = 1.0 \times 10^{-4}$. Every simultaneous $r < 5$ contrast lies below zero. The result is a

broad advantage over the all-initial-search endpoint on this Wall panel, not evidence that $r = 3$ is universally optimal.

6.2 PushT repeats the fixed $r = 4$ versus $r = 5$ direction

On the previously unused 64-episode bank, $r = 4 - r = 5$ is -4.96 (primary 95% episode-block interval $[-7.25, -2.85]$; two-way sensitivity interval $[-7.75, -1.94]$; episode-level sign-flip sensitivity $p = 2.5 \times 10^{-5}$). Physical task cost changes by -4.44, whereas success changes by only $+1.0$ percentage points with an interval containing zero. Thus the observed endpoint difference is numerically concentrated in physical progress, not a separately established success-rate increase. This train-split panel confirms the fixed allocation policy, not held-out visual representation generalization.

The retrospective validation panel has the same aggregate direction in all three maps. Across 21 episodes after map averaging, $r = 4$ records 19 wins, no ties, and 2 losses; $r = 4 - r = 5$ is -11.44 (primary 95% interval $[-18.05, -6.26]$; two-way sensitivity interval $[-18.84, -5.42]$; exact sign-flip sensitivity $p = 4.58 \times 10^{-5}$). Success changes by $+9.5$ percentage points and physical cost by -6.68. Prior diagnostic access makes this a non-blind replication rather than a second confirmation bank.

6.3 A matched Wall information control supports fresh-observation conditioning

Figure 3 summarizes the intervention, matched quantities, and endpoint contrast.

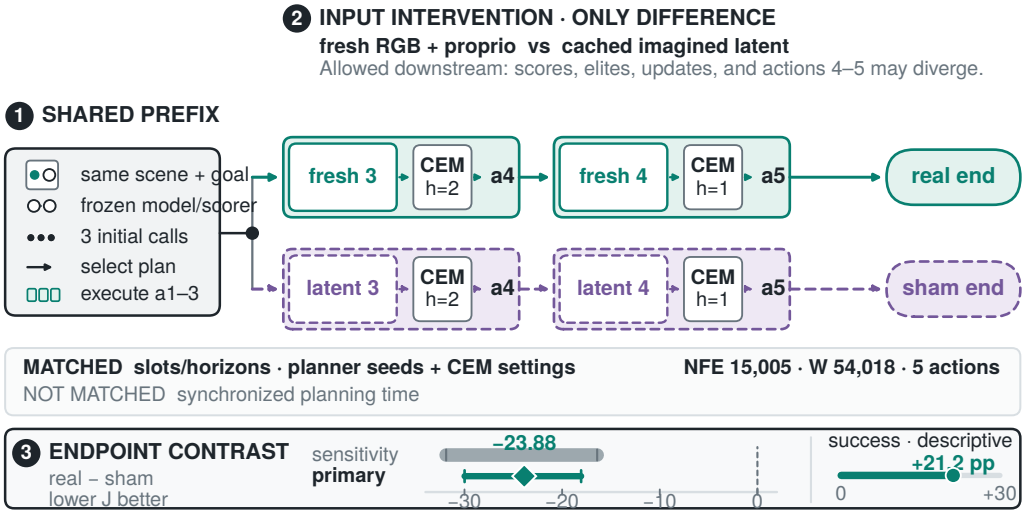


Figure 3: **Matched Wall information control.** Both arms share the scene, goal, frozen model/scorer, three-call realized prefix, later call slots/horizons, planner seeds, CEM settings, five calls, logical NFE, horizon-weighted work, and action count. Only the post-split input differs: fresh RGB/proprioception versus cached imagined latents; subsequent optimizer trajectories may diverge. The endpoint plot shows real-minus-sham cost (diamond mean; dark primary and pale sensitivity 95% intervals) and descriptive success. Elapsed planning time is not matched.

Real feedback minus the cached-latent sham is -23.88 (primary 95% scene-block interval $[-29.99, -18.05]$; two-way sensitivity interval $[-31.88, -16.41]$), with the same sign in all three maps; success changes by $+21.2$ percentage points. Logical NFE and W match exactly, while synchronized planning time is 28.82 versus 26.39 s. This supports fresh-observation conditioning relative to the tested sham, not a decomposition of every change induced by r.

Neither post-hoc PushT development control shows a feedback advantage. At three calls, `commit_2-restart_3` is $+3.06$ (95% interval $[-6.34, 15.66]$); at five calls, `commit_1-restart_5` is $+1.69$

([−6.27, 11.98]). Later calls use shorter horizons, so these match call count rather than physical compute. They do not support a task-general feedback mechanism.

6.4 The Wall direction recurs in targeted adaptations

Table 1 and Figure 4 give within-stack $r = 3 - r = 5$ contrasts on one shared Wall panel. They are adaptation checks, not independent task replications, official end-to-end SOTA reproductions, or cross-stack rankings.

Table 1: Wall within-stack direction checks on one panel. Negative ΔJ favors $r = 3$; success changes are descriptive. iCEM/DecentCEM are adaptations; JEPA uses one official checkpoint.

Stack	ΔJ	Scene-block CI	Two-way CI	Descr. Δ success (pp)
CEM / DINO-WM	-34.71	[-42.78, -26.66]	[-45.03, -24.53]	+31.6
iCEM comp. / DINO-WM	-46.40	[-53.32, -39.44]	[-56.11, -36.49]	+41.7
DecentCEM style / DINO-WM	-49.37	[-57.20, -41.38]	[-58.98, -39.67]	+44.8
CEM / JEPA-WM checkpoint	-4.69	[-8.18, -1.75]	[-9.73, -0.32]	+3.8

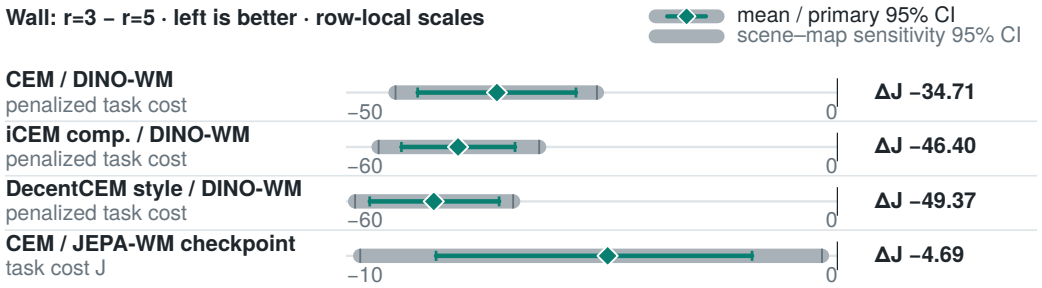


Figure 4: **The Wall direction recurs within four named stacks on one shared panel.** Diamonds are $r = 3 - r = 5$ means; dark/pale bars are primary scene-block/crossed scene–map 95% intervals ($n = 96$ per row). Every interval lies below zero. These are adaptations, not independent task replications or official SOTA reproductions. Scales are row-local; spans must not be compared across rows.

6.5 Resources and adaptive-policy boundaries

Table 2 reports the resource deltas for the registered contrasts. Logical NFE and batched rollout invocations match within each row, but schedules that reserve later calls use less horizon-weighted work and synchronized time. Peak memory is stable within each pair. Equal logical NFE is therefore not equal physical compute; Appendix Table 8 gives every arm separately.

Table 2: Registered resource contrasts (selected schedule minus $r = 5$). Logical NFE and batched rollout invocations match within each pair; Appendix Table 8 reports per-arm batch counts. Negative ΔW and Δtime mean lower use. Memory is maximum allocated/reserved GiB and is equal within each pair.

Panel	NFE/arm	ΔW	Δtime (s)	CUDA GiB
PointMaze $r = 1 - r = 5$	15,005	-30,010	-52.25	18.26/24.72
Wall $r = 3 - r = 5$	15,005	-21,007	-8.86	4.31/6.85
PushT confirmation $r = 4 - r = 5$	45,005	-36,004	-58.33	18.01/23.86
PushT replication $r = 4 - r = 5$	45,005	-36,004	-58.31	18.01/23.86

Adaptive variants do not support a stronger claim. The AdaReP-inspired Wall controller reduces mean calls from 5.00 to 3.99 and W from 54,018 to 36,596, but costs 16.30 versus 13.98 for fixed $r = 3$ and fails its quality gates. On its 39-instance panel, the sealed selector costs 22.96 versus 22.70 for fixed $r = 4$ (difference $+0.26$, 95% interval $[-2.15, 2.69]$). A safe-binary selector improves over $r = 4$ on its confirmation panel (-1.01, $[-2.23, -0.10]$) but is worse than fixed $r = 1, 2, 3$, so its all-fixed superiority gate fails. These results delimit the paper to precommitted schedules.

7 Limitations and Discussion

Task and horizon scope. The evidence covers $K = H = 5$, three tasks, and frozen visual-world-model controllers. In PointMaze, the selected contrast and every simultaneous $r < 5$ contrast include zero; Wall and PushT therefore do not imply a generally optimal replanning frequency. Other budgets, horizons, stochastic dynamics, and observation regimes remain open.

Identification and compute. Changing r jointly changes initial search breadth, commitment, observation access, later horizons, and W. The Wall sham isolates a narrower information intervention relative to one cached-latent control, but arm-specific scores can induce different optimizer trajectories; the post-hoc PushT controls do not support a task-general feedback mechanism. Equal calls and logical NFE do not equalize work or time, and energy and monetary cost were not measured.

Model, randomness, and split coverage. The main panels use task-specific DINO-WM checkpoints; the second-model check is one JEPA-WM Wall checkpoint. No official end-to-end external SOTA reproduction is included: all four robustness rows reuse one Wall panel, and iCEM/DecentCEM are adaptations. Three seed maps provide sensitivity, not generalization to arbitrary optimizer randomness. PushT confirmation uses the model-training split; validation is retrospective and non-blind, so neither establishes held-out representation generalization.

Adaptive and ranking boundaries. The AdaReP-inspired monitor trades quality for fewer calls, the sealed selector does not beat its best fixed comparator, and the safe-binary selector fails the all-fixed gate. Planning-error diagnostics reveal weak latent ranking, but the tested repairs do not correct it. We therefore claim neither adaptive superiority nor action-ranking repair: temporal placement should be specified and measured as part of a frozen-controller protocol.

8 Conclusion

With five complete planner calls and matched logical NFE, their precommitted temporal placement changes performance in some, but not all, frozen visual-world-model protocols. Wall provides the strongest evidence, including a matched cached-latent sham; PushT repeats one fixed $r = 4$ versus $r = 5$ direction, while PointMaze and the adaptive/equal-call controls bound broader interpretations. Planner-call timing is therefore a measurable, task-dependent controller choice—not a universal replanning rule or a substitute for validating the model and scorer themselves.

AI use statement

In this work, generative AI tools were used only (i) to aid or polish writing; (ii) for retrieval and discovery, including finding related work; and (iii) to draft sections of the paper. The authors reviewed and verified all AI-assisted material and take full responsibility for the final content.

Reproducibility statement

Appendices A–F document panels, units, statistics, results, resources, controls, diagnostics, and lineage. The anonymous supplement provides deterministic figure code, reviewed outputs, redacted machine-readable result/claim contracts, and integrity manifests. End-to-end experiment code, pretrained checkpoints, upstream task assets, and raw per-unit trajectories are not redistributed.

REFERENCES

Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. In *International Conference on Learning Representations*, 2021. URL https://iclr.cc/virtual/2021/poster/3240.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Homanga Bharadhwaj, Kevin Xie, and Florian Shkurti. Model-predictive control via Cross-Entropy and gradient-based optimization. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 277–286. PMLR, 2020. URL https://proceedings.mlr.press/v120/bharadhwaj20a.html.

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

Yutian Cheng, Xiaojian Ma, Xianhao Wang, Min Yang, Rongpeng Su, Hangxin Liu, Xi Chen, Shuai Li, and Qing Li. AdaReP: Adaptive re-planning under model mismatch for neural world-model predictive control. In *Artificial Neural Networks and Machine Learning – ICANN 2026*, volume 17093 of *Lecture Notes in Computer Science*, pp. 216–228. Springer Nature Switzerland, 2027. doi: 10.1007/978-3-032-38404-1_18. URL https://doi.org/10.1007/978-3-032-38404-1_18.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://www.roboticsproceedings.org/rss19/p026.html.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, pp. 530–540, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

Junha Chun, Youngjoon Jeong, and Taesup Kim. Sparse imagination for efficient visual world model planning. In *International Conference on Learning Representations*, pp. 102160–102188, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a750d52284ff70c6d6bab8072c392d74-Abstract-Conference.html.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the Cross-Entropy method. *Annals of Operations Research*, 134(1):19–67, 2005. doi: 10.1007/s10479-005-5724-z. URL https://doi.org/10.1007/s10479-005-5724-z.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation*, pp. 2786–2793. IEEE, 2017. doi: 10.1109/ICRA.2017.7989324. URL https://doi.org/10.1109/ICRA.2017.7989324.

Christopher Grimm, André Barreto, Satinder P. Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5541–5552. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/3bb585ea00014b0e3ebe4c6dd165a358-Abstract.html.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Nicklas A. Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8387–8406. PMLR, 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Xiaodi Huang, Ziyi Ding, Jingtian Wan, Yuchen Liu, Yuan Zhang, Xiao-Ping Zhang, Jiayu Chen, Zhang Zhang, and Tao Huang. Traj-LeWM: Path-aware world-model planning via latent trajectory cost, 2026. URL https://arxiv.org/abs/2608.14125.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9902–9915. PMLR, 2022. URL https://proceedings.mlr.press/v162/janner22a.html.

Huiping Li and Yang Shi. Event-triggered robust model predictive control of continuous-time nonlinear systems. *Automatica*, 50(5):1507–1513, 2014. doi: 10.1016/j.automatica.2014.03.015. URL https://doi.org/10.1016/j.automatica.2014.03.015.

Liangyu Li, Shengzhi Wang, Libin Qiu, Mingliang Xiong, and Qingwen Liu. World model control by trajectory reachability metrics, 2026. URL https://arxiv.org/abs/2605.22164.

Yuanchang Liang, Xiaobo Wang, Kai Wang, Shuo Wang, Xiaojiang Peng, Haoyu Chen, David Kim Huat Chua, and Prahlad Vadakkepat. Adaptive action chunking at inference-time for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20802–20811, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liang_Adaptive_Action_Chunking_at_Inference-time_for_Vision-Language-Action_Models_CVPR_2026_paper.html.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorld-Model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026. URL https://arxiv.org/abs/2603.19312.

Yi Pan, Miao Pan, Qi Lu, Jiaming Huang, Man Zhang, Siteng Huang, Xin Li, Jie Zhang, Yongliang Shen, Xuhong Zhang, and Wenqi Zhang. VLA-Corrector: Lightweight detect-and-correct inference for adaptive action horizon, 2026. URL https://arxiv.org/abs/2607.01804.

Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient Cross-Entropy method for real-time planning. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 1049–1065. PMLR, 2021. URL https://proceedings.mlr.press/v155/pinneri21a.html.

Michael Psenka, Michael Rabbat, Aditi Krishnapriyan, Yann LeCun, and Amir Bar. Parallel stochastic gradient-based planning for world models, 2026. URL https://arxiv.org/abs/2602.00475.

Jacob Sacks and Byron Boots. Learning sampling distributions for model predictive control. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1733–1742. PMLR, 2023. URL https://proceedings.mlr.press/v205/sacks23a.html.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1332–1344. PMLR, 2023. URL https://proceedings.mlr.press/v205/seo23a.html.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. *arXiv preprint arXiv:2502.14819*, 2025. URL https://arxiv.org/abs/2502.14819.

Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *Transactions on Machine Learning Research*, May 2026. URL https://openreview.net/forum?id=cHZn5Gdh8e. Reproducibility certification.

Rui Wang, Yue Zhang, Jiehong Lin, Kuncheng Luo, Jianan Wang, Zhongrui Wang, and Xiaojuan Qi. When to trust imagination: Adaptive action execution for world action models, 2026a. URL https://arxiv.org/abs/2605.06222.

Tingwu Wang and Jimmy Ba. Exploring model-based planning with policy networks. In *International Conference on Learning Representations*, 2020. URL https://iclr.cc/virtual/2020/poster/1552.

Ying Wang, Oumayma Bounou, Yann LeCun, and Mengye Ren. AdaJEPA: An adaptive latent world model, 2026b. URL https://arxiv.org/abs/2606.32026.

Zihua Wang, Zhitao Lin, Ruibo Li, Yu Zhang, Xu Yang, Siya Mi, and Xiu-Shen Wei. Open-loop planning, closed-loop verification: Speculative verification for VLA, 2026c. URL https://arxiv.org/abs/2604.02965.

William F. Whitney, Jake Varley, Deepali Jain, Krzysztof Marcin Choromanski, Sumeet Singh, and Vikas Sindhwani. Modeling the real world with high-density visual particle dynamics. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1427–1442. PMLR, 2025. URL https://proceedings.mlr.press/v270/whitney25a.html.

Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2): 344–357, 2017. doi: 10.2514/1.G001921. URL https://doi.org/10.2514/1.G001921.

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, 2026. URL https://arxiv.org/abs/2608.03483.

Feng Ye, Yiming Zhao, Yong Yu, Hongxu Zhou, Yong Pan, Yuan Xue, Peng Jia, and Chuanmin Jia. Rethink before you execute: Adaptive execution for world action models, 2026. URL https://arxiv.org/abs/2608.09492.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M. Patel, Paul Pu Liang, Daniel Khashabi, Cheng Peng, Rama Chellappa, Tianmin Shu, Alan Yuille, Yilun Du, and Jieneng Chen. World-in-world: World models in a closed-loop world. In *International Conference on Learning Representations*, pp. 55660–55699, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5b4263be85820683d78675cc18d2efc7-Abstract-Conference.html. Oral presentation.

Wancong Zhang, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, and Nicolas Ballas. Hierarchical planning with latent world models, 2026b. URL https://arxiv.org/abs/2604.03208.

Zhengshu Zhang and Zhiyuan Li. ARC-Bench: Closed-loop replanning masks broken action ranking in frozen JEPA world models, 2026. URL https://arxiv.org/abs/2609.05461.

Zichen Zhang, Jun Jin, Martin Jagersand, Jun Luo, and Dale Schuurmans. A simple decentralized Cross-Entropy method. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36495–36506, 2022. doi: 10.52202/068431-2644. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/ec9dc93250548578aa4569aa19acfd81-Abstract-Conference.html.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016. URL https://www.roboticsproceedings.org/rss19/p016.html.

Wentao Zhao, Jiaming Chen, Ziyu Meng, Donghui Mao, Ran Song, and Wei Zhang. VLMPC: Vision-language model predictive control for robotic manipulation. In *Proceedings of Robotics: Science and Systems XX*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.106. URL https://www.roboticsproceedings.org/rss20/p106.html. Paper 106, 13 pages.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhou25t.html.

Siyuan Zhou, Yilun Du, Shun Zhang, Mengdi Xu, Yikang Shen, Wei Xiao, Dit-Yan Yeung, and Chuang Gan. Adaptive online replanning with diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44000–44016, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/893a5db6100028ec814cfd99fe92c31b-Abstract-Conference.html.

APPENDIX

A EVALUATION PROTOCOL AND STATISTICAL UNITS

A.1 LOCKED PANELS AND EVIDENTIAL ROLES

Table 3 separates development, confirmation, robustness, and diagnostic panels. "Cells" are descriptive scene/episode–map evaluations; the inferential unit is the scene or episode after averaging the three registered maps.

Table 3: Locked evaluation panels. No full or confirmation outcome revised a pilot choice, model, scorer, or analysis rule.

Panel	Unit / size	Maps	Status	Evidential role
PointMaze pilot	scene / 32	3	development	Select $r = 1$ by mean endpoint; disjoint from full set.
PointMaze full	scene / 96	3	registered	Primary $r = 1 - r = 5$ contrast and complete five-point curve.
Wall pilot	scene / 32	3	development	Select $r = 3$ by mean endpoint; disjoint from full set.
Wall full	scene / 96	3	registered	Primary $r = 3 - r = 5$ contrast and complete five-point curve.
Wall matched sham	scene / 96	3	registered control	Fresh RGB+proprio versus cached imagined latent at matched later slots, horizons, logical NFE, and W.
Wall iCEM / DecentCEM	scene / 96	3	targeted robustness	Within-adaptation $r = 3 - r = 5$ direction; not official reproductions.
Wall JEPA-WM	scene / 96	3	targeted transfer	One official checkpoint in a protocol-matched project evaluator.
PushT development	episode / 124	1	development	Select $r = 4$; all fixed schedules and oracle reported.
PushT confirmation	episode / 64	3	prospective policy test	Previously unused train-split bank; not a representation holdout.
PushT validation	episode / 21	3	retrospective	Non-blind replication after historical diagnostic access.
PushT equal-call controls	episode / 24	1	post hoc	Test feedback against independent full-horizon restarts at 3 and 5 calls.
Adaptive / ranking branches	reset / 8–96	varies	development or sealed	Bound adaptive and ranking-repair claims; panel-specific status reported below.

A.2 INFERENCE DETAILS

For scene/episode-block intervals, let $\bar{\delta}_e = L^{-1} \sum_{m=1}^{L} [J_{e,m}(s_1) - J_{e,m}(s_0)]$ with $L = 3$ registered planner-seed maps. We resample the n scenes or episodes with replacement, compute the mean $\bar{\delta}$ in each of 20,000 draws, and use percentile endpoints. The crossed sensitivity bootstrap independently resamples scenes/episodes and the three map labels before recomputing the mean; with only three maps, it is a sensitivity analysis rather than precise seed-population inference.

For PointMaze and Wall, the two-sided sign-flip sensitivity repeatedly multiplies each $\bar{\delta}_e$ by an independent Rademacher sign, computes the absolute paired mean, and uses 20,000 draws with the $(b+1)/(B+1)$ correction; the two task values are Holm adjusted. The 64-episode PushT panel uses the same episode-level statistic and 200,000 Monte Carlo sign draws with the frozen analysis seed. The 21-episode PushT panel enumerates all 2^{21} sign patterns exactly. None is interpreted as a treatment-randomization test. Success is secondary; reported changes remain descriptive unless an interval or test is given.

B Complete fixed-schedule results

B.1 PointMaze and Wall five-point curves

Table 4: Complete standard-CEM curves. Each row contains 96 scenes crossed with three registered planner-seed maps (288 descriptive cells). Lower J and distance are better. Logical NFE is 15,005 in every row.

Task	r	Mean J	Success	Mean distance	Planning (s)	W
PointMaze	1	14.8084	259/288	4.7390	57.5048	45,015
PointMaze	2	14.8953	259/288	4.8258	62.9691	48,016
PointMaze	3	15.2764	258/288	4.8597	73.8927	54,018
PointMaze	4	15.1033	259/288	5.0338	90.2771	63,021
PointMaze	5	16.3718	255/288	4.9134	109.7520	75,025
Wall	1	11.8439	260/288	2.1216	25.0640	45,015
Wall	2	9.2825	266/288	1.6436	26.3321	48,016
Wall	3	10.5128	263/288	1.8323	28.8609	54,018
Wall	4	9.4177	266/288	1.7788	32.6541	63,021
Wall	5	45.2257	172/288	4.9479	37.7168	75,025

The Wall full-set minimum is $r = 2$, followed by $r = 4$; both are post-hoc descriptions. The registered primary remains pilot-selected $r = 3$ versus $r = 5$. Table 5 reports all primary uncertainty and map-wise signs.

Table 5: Registered PointMaze/Wall contrasts. W/T/L counts are over the 288 descriptive cells and are not substituted for scene-level inference.

Task	Contrast	Mean ΔJ	Scene 95% CI	Two-way 95% CI
PointMaze	$r = 1 - r = 5$	-1.5633	[-3.5876, 0.0923]	[-4.3285, 0.7295]
Wall	$r = 3 - r = 5$	-34.7128	[-42.7817, -26.6626]	[-45.0337, -24.5300]

Task	Map 1 mean	Map 2 mean	Map 3 mean	W/T/L
PointMaze	-1.120	-2.307	-1.263	160/0/128
Wall	-41.119	-32.144	-30.875	264/2/22

B.2 PushT development, confirmation, and replication

Table 6: Complete 24-reset PushT development curve used to select $r = 4$. All rows have 45,005 logical candidate trajectories; they are development results, not confirmatory evidence.

Schedule	Mean J	Mean distance	Success	Planning (s)
$r = 1$	19.5882	15.4216	0.9167	177.6945
$r = 2$	22.0922	15.8422	0.8750	194.9708
$r = 3$	16.2420	14.1587	0.9583	228.7989
$r = 4$	12.9922	12.9922	1.0000	279.4770
$r = 5$	17.8952	17.8952	1.0000	339.5585
Hindsight oracle (non-deployable)	8.1301	—	—	—

Table 7: Locked PushT arm summaries. Counts are over descriptive episode–map cells; inference averages maps within each episode.

Panel	Arm	Mean J	Mean distance	Success	Time (s)	Logical NFE	W
64-episode confirmation	$r = 4$	20.4355	16.2688	176/192	270.4419	45,005	189,021
64-episode confirmation	$r = 5$	25.4000	20.7125	174/192	328.7684	45,005	225,025
21-episode replication	$r = 4$	12.2546	12.2546	63/63	270.3957	45,005	189,021
21-episode replication	$r = 5$	23.6986	18.9367	57/63	328.7065	45,005	225,025

C Resource accounting

Table 8: Arm-level resource ledger for the registered comparisons. Memory is the maximum observed CUDA allocated/reserved GiB. Equal logical NFE does not equalize W or time.

Panel	Arm	Calls	Logical NFE	Batched rollouts	W	Time (s)	CUDA GiB
PointMaze	$r = 1$	5	15,005	55	45,015	57.50	18.26/24.72
PointMaze	$r = 5$	5	15,005	55	75,025	109.75	18.26/24.72
Wall	$r = 3$	5	15,005	55	54,018	28.86	4.31/6.85
Wall	$r = 5$	5	15,005	55	75,025	37.72	4.31/6.85
PushT confirmation	$r = 4$	5	45,005	155	189,021	270.44	18.01/23.86
PushT confirmation	$r = 5$	5	45,005	155	225,025	328.77	18.01/23.86
PushT replication	$r = 4$	5	45,005	155	189,021	270.40	18.01/23.86
PushT replication	$r = 5$	5	45,005	155	225,025	328.71	18.01/23.86

Table 9: Wall robustness-resource summaries. Each row compares $r = 3 - r = 5$ within one stack. The two arms have equal maximum allocated/reserved memory within a row.

Stack	NFE/arm	ΔW	Δ time (s)	Alloc./reserv. GiB
CEM / DINO-WM	15,005	-21,007	-8.856	4.31/6.85
iCEM comp. / DINO-WM	15,000	-21,000	-8.815	4.31/6.85
DecentCEM style / DINO-WM	15,005	-21,007	-8.816	3.78/6.16
CEM / JEPA-WM checkpoint	15,005	-21,007	-27.363	15.40/16.83

Energy and monetary cost were not logged and are not inferred from time or memory. Shared call materialization reduced collection duplication but is not credited as an arm-level speedup.

D POST-HOC CONTROLS AND ADAPTIVE POLICIES

D.1 PUSHT FULL-HORIZON RESTART CONTROLS

Table 10: All 24-reset PushT compute-control methods. `commit` schedules use feedback; `restart` schedules repeat independent full-horizon searches.

Method	Mean J	Mean distance	Success	Planning (s)
open loop (1 call)	28.1641	23.9974	0.9167	67.7232
`commit_2` (3 calls)	25.1610	18.9110	0.8750	109.4690
`restart_2` (2 calls)	24.8747	20.7080	0.9167	135.7255
`restart_3` (3 calls)	22.0981	20.0148	0.9583	203.6760
`commit_1` (5 calls)	19.5882	15.4216	0.9167	177.6945
`restart_5` (5 calls)	17.8952	17.8952	1.0000	339.5585

At three calls, `commit_2-restart_3` is $+3.0629$ (95% CI $[-6.3428, 15.6568]$, sign-flip $p = 0.6223$). At five calls, `commit_1-restart_5` is $+1.6931$ ($[-6.2692, 11.9833]$, $p = 0.7222$). Both intervals include zero and both point estimates disfavor feedback. The additional `restart_2` arm is descriptive: `commit_2-restart_2` is $+0.2863$ ($[-7.0098, 11.4212]$, $p = 0.9940$), but this is a three-call versus two-call comparison and is not an equal-call test.

D.2 SINGLE-CALL CONTROLS

The two post-hoc single-CEM controls use 45,001 trajectories versus 45,005 for the five-call references, but they differ in batched invocations, W, time, and memory. Table 11 therefore treats them as robustness checks rather than exact physical-compute matches.

Table 11: Single-CEM controls and references on the 39-instance selector-final panel.

Method	Mean J	Success	Traj.	W	Batches	Time (s)
single deep (300×150)	35.1532	0.8205	45,001	225,005	151	328.62
single balanced (450×100)	30.3554	0.8718	45,001	225,005	101	332.64
fixed $r = 5$	28.2473	0.9231	45,005	225,025	155	329.75
fixed $r = 4$	22.6970	0.9487	45,005	189,021	155	271.26
sealed selector	22.9570	0.9487	45,005	156,941	155	212.55

Balanced minus fixed $r = 4$ is $+7.6584$ (95% CI $[0.3728, 14.9755]$); deep minus fixed $r = 4$ is $+12.4563$ ($[4.0968, 21.0651]$). Median allocated/reserved memory is 26.89/35.59 GiB for balanced and 18.01/23.86 GiB for deep, reinforcing that equal trajectory counts do not imply equal resource profiles.

D.3 SEALED SELECTOR, SAFE-BINARY SELECTOR, AND ADAPTIVE WALL CONTROLLER

Table 12: Sealed PushT selector-final panel ($n = 39$). The oracle is non-deployable and shown only as headroom.

Policy	Mean J	Success	Mean r	Mean W	Selection counts $r = 1{:}5$
fixed $r = 1$	26.9322	0.8974	1.000	135,015	39/0/0/0/0
fixed $r = 2$	25.6070	0.9231	2.000	144,016	0/39/0/0/0
fixed $r = 3$	30.1502	0.8462	3.000	162,018	0/0/39/0/0
fixed $r = 4$	22.6970	0.9487	4.000	189,021	0/0/0/39/0
fixed $r = 5$	28.2473	0.9231	5.000	225,025	0/0/0/0/39
sealed selector	22.9570	0.9487	2.487	156,941	8/13/10/7/1
hindsight oracle	16.8304	0.9487	2.974	168,019	6/7/12/10/4

The sealed selector minus pre-registered fixed $r = 2$ is -2.6501 (95% CI $[-7.0211, 0.6416]$, $p = 0.2661$), but versus the post-hoc best fixed $r = 4$ it is $+0.2600$ ($[-2.1506, 2.6883]$, $p = 0.8385$). The safe-binary confirmation policy selects $r = 3$ for 15 episodes and $r = 4$ for 49; it improves on fixed $r = 4$ by -1.0120 ($[-2.2253, -0.0994]$, $p = 0.04665$) with no success-count loss, yet costs 20.0509 versus 16.6661/16.1287/17.0456 for fixed $r = 1/2/3$. Its pre-specified all-fixed superiority gate is therefore false.

On Wall, the AdaReP-inspired controller averages 3.9896 calls, NFE 11,972.7, $W = 36{,}595.5$, and 20.2578 s, versus five calls, NFE 15,005, $W = 54{,}018$, and 28.7809 s for fixed $r = 3$. The endpoint difference is unfavorable ($+2.3245$, two-way 95% CI $[-3.9399, 8.7798]$) and success differs by -1.74 percentage points ($[-6.94, 3.47]$). It demonstrates an efficiency–quality trade-off, not Pareto superiority.

D.4 Regret certificates

On a 24-reset development panel, an always-switch relative-regret policy changes cost by -0.4447 with interval $[-7.2269, 6.3353]$ and makes 12 harmful overrides among 24. The certified policy makes zero switches; marginal coverage is 0.8333 against a 0.9 target. A separate adaptive-regret certificate is likewise silent at $\alpha = 0.1/0.2$; at $\alpha = 0.3$ it makes one harmful switch and changes cost by $+2.2042$ ($[0, 6.6127]$). Neither branch advanced.

E Planning diagnostics and attempted repairs

A post-hoc eight-reset planning-error audit finds mean physical regret 8.4967 between the returned CEM mean and the best physical elite, and mean 9.3977 for the best predicted elite relative to the physical oracle. The elite predicted–physical Spearman correlation is 0.0785 and physical ranking disagreement is 0.4695. Every pool contains a successful elite, but the best-predicted elite succeeds on 7/8 resets. These development diagnostics motivate caution about latent action ranking; they do not identify a population effect.

Table 13: Direct factual fine-tuning results on 16 development resets. Positive Δ physical cost is worse. Every registered 200-step variant degrades the frozen original.

Variant	Physical cost	Success	Δ vs. original	95% CI
Frozen original	18.7488	0.9375	0	[0,0]
Dynamics only	35.0101	0.2500	+16.2612	[10.9159,22.0223]
Factual rank	37.1984	0.4375	+18.4496	[11.3023,25.3881]
Elite corrected	40.9528	0.3125	+22.2040	[12.2382,34.9083]
Factual calibrated	41.6157	0.3125	+22.8668	[15.8769,30.0953]

Early checkpoints do not rescue this branch: all 50/100/200-step contrasts are positive; the smallest is elite-corrected at 100 steps ($+14.2373$, 95% CI $[7.6119, 20.6647]$). A factual risk head changes cost by $+0.0369$ ($[-2.8121, 2.7346]$) and reduces success from 1.0 to 0.875 on eight resets. A task critic switches two of 24 decisions and changes cost by $+0.3600$ ($[0, 1.0757]$). Shadow penalties and trust-region interpolation also fail advancement; the closest interpolation has cost 19.8599 versus 18.7488 for the frozen baseline. These are valid negative results and explain why the paper makes no ranking-repair claim.

F COMPLETE EXPERIMENTAL LINEAGE

Table 14 lists all 40 tracked objects from the frozen experiment audit. Scientific negatives are retained above; engineering failures are marked invalid and contribute no effect estimate, sample size, or failure rate.

Table 14: Experiment lineage and paper disposition. "Support" denotes data/provenance generation, not an independent performance result.

#	Tracked object	Final status	Scientific use
1	`adaptive_regret_collection`	support complete	Training-data provenance only.
2	`adaptive_regret_pilot`	negative	Certificate silent or harmful; no advancement.
3	`assets`	support complete	Asset provenance only.
4	`core_assets`	support complete	Model/data asset provenance only.
5	`cross_task_allocation`	completed	Single-map precursor; superseded by multi-map statistics.
6	`cross_task_data`	support complete	Official data provenance.
7	`cross_task_planner_seed_replication`	authoritative	Main PointMaze/Wall curves and registered contrasts.
8	`equal_nfe_cem_control`	post-hoc	Single-CEM robustness; not exact physical-compute matching.
9	`experiment_phase_lock_20260923`	governance complete	Freezes authoritative artifacts and stop rule.
10	`factual_decision_collection`	support complete	Critic-training data provenance.
11	`factual_evaluation`	negative	Direct factual fine-tuning systematically degrades planning.
12	`factual_evaluation_early`	negative	Early checkpoints do not rescue factual tuning.
13	`factual_pilot`	support complete	Training provenance; initial shape error invalid.
14	`factual_risk_head_pilot`	negative	Development benefit does not transfer; not advanced.
15	`factual_task_critic`	negative	Two switches, no execution benefit.
16	`feedback_compute_control`	negative	PushT equal-call feedback gate is false.
17	`final_zero_gpu_risk_audit_20260924`	governance complete	Hash, replay, process, and stop-rule audit.
18	`landscape_preserving_pilot`	negative	Shadow/interpolation variants do not advance.
19	`planning_error_audit`	diagnostic	Small development panel exposes ranking error.

Continued on next page

Table 14: Experiment lineage and paper disposition (continued).

#	Tracked object	Final status	Scientific use
20	`pusht_heldout_r4r5_v1`	authoritative	Retrospective, non-blind validation-split replication.
21	`pusht_r4r5_confirmation_v1`	authoritative	Prospective train-split allocation-policy confirmation.
22	`regret_certificate_collection`	support complete	Certificate-data provenance; first run invalid.
23	`regret_certificate_pilot`	negative	Coverage misses target; certified policy is silent.
24	`runtime`	smoke only	Connectivity/runtime smoke, no scientific inference.
25	`search_feedback_allocation`	development	Selects PushT $r = 4$; full curve reported above.
26	`search_feedback_selector`	development holdout	Authorizes one sealed test; not a headline result.
27	`search_feedback_selector_final`	sealed mixed	Favors pre-registered $r = 2$ in mean, not best fixed $r = 4$.
28	`search_feedback_selector_v2`	negative development	Stable learned signal not found; no new-bank access.
29	`selector_safe_binary_v1`	mixed	Beats $r = 4$ locally, fails all-fixed superiority.
30	`selector_safe_binary_v1_development`	historical	Candidate-freezing provenance only.
31	`temporal_commitment_ceiling`	positive but confounded	More calls improve over one-call open loop; compute differs.
32	`wall_adarep_inspired_v1`	mixed	Saves calls/work/time but fails quality gates.
33	`wall_decentcem_schedule_v1`	authoritative	Within-adaptation Wall direction check.
34	`wall_icem_schedule_v1`	invalid	No valid full rows; engineering-only record.
35	`wall_icem_schedule_v2`	invalid	Smoke only; engineering-only record.
36	`wall_icem_schedule_v3`	invalid partial	Partial rows prohibited from scientific inference.
37	`wall_icem_schedule_v4`	authoritative	Final iCEM-component Wall direction check.
38	`wall_jepawm_cem_schedule_v1`	authoritative	One-checkpoint JEPA-WM direction check.
39	`wall_matched_feedback_sham_v2`	authoritative	Matched fresh-observation information control.
40	`wall_matched_feedback_sham_v1`	invalid pre-evaluation	No evaluation output and no scientific use.