SURVEYOR: TRAINING-FREE DECOMPOSITION FOR LONG-RANGE LATENT WORLD-MODEL PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent world-model planners succeed at short goal distances but fail at long ones, and hierarchical variants repair this failure by decomposing a distant goal into a sequence of latent subgoals. However, existing hierarchical planners make two decisions by fixed schedule rather than by evidence: they regenerate the subgoal sequence at every replanning step, even when the previous draft remains consistent with execution, and they decompose every goal, even when the executor could reach it directly. We introduce SURVEYOR, a training-free, plug-and-play decomposition layer that sits between an executor and a frozen world model and makes both decisions from the world model's own latents and predictions. A *drafter* proposes latent subgoals a fixed number of steps ahead; an *accept rule* reuses the remaining draft while the achieved latent state agrees with the subgoal just pursued; and an *arbiter* bypasses or retires drafting once the goal is predicted to be within the executor's reach. Both decisions use a tolerance calibrated offline from observation pairs that the benchmark's success criterion treats as equivalent. Extensive experiments over five executors, spanning sampling-based, gradient-based and amortized planners, and four environments show that SURVEYOR raises success at the longest PushT goal distance from below 10% to above 92% for every search-based executor while issuing up to two fifths fewer drafter calls than every-step drafting, and improves both the average and the worst-case success of all five executors over the seventeen combinations of environment and goal distance.

1 INTRODUCTION

World models let agents anticipate the consequences of their actions and improve behavior through imagined interaction (Ha & Schmidhuber, 2018; Hafner et al., 2023; Hansen et al., 2024). Latent world models make this tractable by planning in a learned representation space: a frozen encoder maps observations to latents, a dynamics model predicts how they evolve, and a controller optimizes actions against imagined futures (Maes et al., 2026b; Zhou et al., 2024; Sobal et al., 2026). Hierarchical variants extend this paradigm to long-horizon tasks by decomposing a distant goal into a sequence of latent subgoals within the horizon of a short-horizon controller (Masip et al., 2026; Janner et al., 2022).

The controller that consumes those subgoals, which we call the *executor*, is the part of the stack whose success degrades with goal distance: as Figure 1 shows, a sampling or gradient-based search over the world model solves nearly every PushT episode whose goal lies 25 steps away and almost none whose goal lies 150 steps away, and an amortized goal-conditioned policy degrades more gracefully along the same axis. Decomposition repairs this failure, but it raises two questions that existing hierarchical planners answer by fixed schedules rather than by evidence. First, a subgoal sequence regenerated at every replanning step costs one generative call per step, whether or not the previous draft still describes what the executor is doing (Masip et al., 2026; Janner et al., 2022; Chen et al., 2024). Second, an executor that can already reach the goal gains nothing from subgoals, and at short goal distances subgoals can lower its success. An adaptive planner should therefore decide when an existing draft can be reused and when decomposition is needed at all.

We propose to make both decisions from the world model's own latents and predictions. At each replanning boundary the planner can compare the latent of the observation it reached with the waypoint it pursued: agreement motivates reuse of the remaining draft, deviation motivates re-

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.

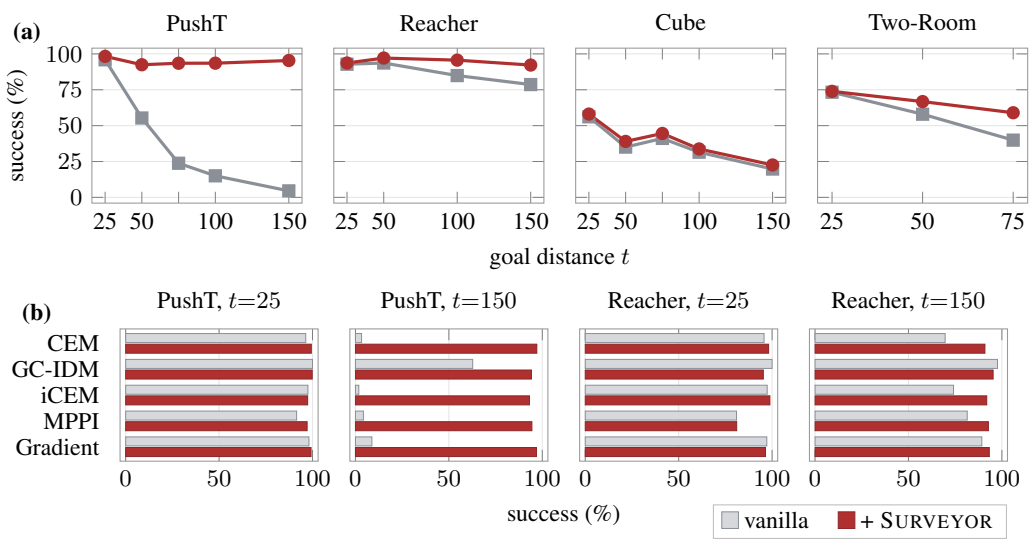

Figure 1: **Success against goal distance, and at the two ends of the range. (a)** Mean success of the four search executors: on PushT and Reacher SURVEYOR holds its success as the goal recedes while the vanilla executors fall; on Cube the gain is small, and on Two-Room it matches the vanilla executors at t=25 and gains beyond. **(b)** Every executor at the two ends of PushT and Reacher. Means over eight repeated runs; Appendix A gives every value.

drafting from the achieved state. At the same boundary a flat plan toward the goal yields a predicted terminal latent whose discrepancy from the goal estimates whether decomposition is still required. Both discrepancies are read against a tolerance calibrated offline from observation pairs that the benchmark's success criterion treats as equivalent.

Building on this principle, we introduce SURVEYOR, a training-free decomposition layer that sits between an executor and the world model, as Figure 2 illustrates. A *drafter* proposes latent subgoals a fixed number of steps ahead; an *accept rule* reuses the remaining draft while the execution–plan consistency check holds and re-drafts when it fails; and an *arbiter* skips drafting when a flat plan is predicted to reach the goal and retires it once that prediction holds. The layer changes no executor and trains nothing, so we place it unchanged over sampling solvers, a gradient solver and an amortized policy and evaluate it by adding one component at a time.

Our contributions are fourfold.

- We show that decomposition lifts every executor we test at the longest PushT distance, where success rises from below 10% to above 92% over the sampling solvers CEM, iCEM and MPPI and a gradient solver, and from 62.7% to 94.2% over the amortized planner GC-IDM (Section 4.2).
- We introduce execution–plan consistency as an online criterion for reusing a drafted subgoal plan; the resulting accept rule keeps that success at 0.58 to 0.73 of the drafter calls of every-step drafting (Section 4.3).
- We introduce an arbiter that reads predicted goal reachability against a tolerance calibrated in the same way; it recovers most of the short-range success that decomposition gives up, raising the worst success rate over the PushT and Reacher settings from at most 63.6% under the accept rule alone to at least 81.0% on every executor (Section 4.4).
- We evaluate the layer unchanged over five executors and four environments, report where it fails, and release the code and the evaluation episodes behind every table.

2 RELATED WORK

LeWM (Maes et al., 2026b) plans with CEM (De Boer et al., 2005) over a frozen JEPA latent space, DINO-WM (Zhou et al., 2024) and PLDM (Sobal et al., 2026) take the same frozen-encoder route,

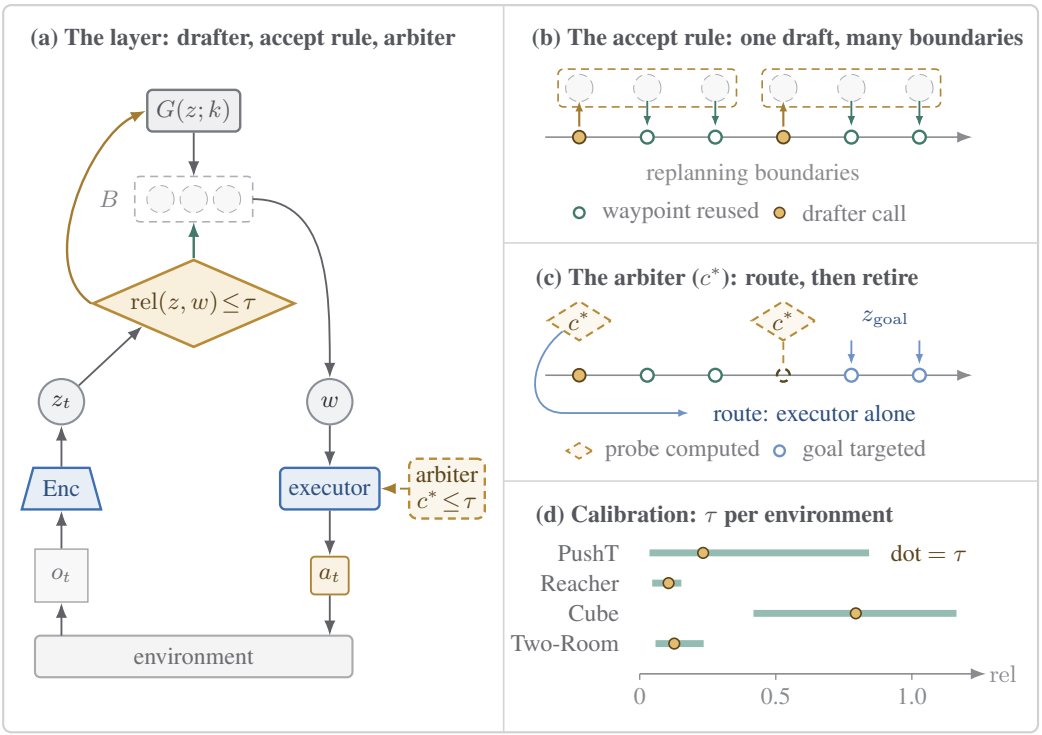

Figure 2: **Overview of SURVEYOR.** Colour marks provenance: the frozen world-model stack (blue), the drafter G (grey) and the layer's additions (amber). **(a)** At each replanning boundary the accept rule compares the achieved latent z_t with the waypoint w just pursued; within the tolerance τ the next waypoint of the block B is used at no drafter cost (teal), and otherwise B is re-drafted from the achieved state (amber). **(b)** One draft of N waypoints is consumed across several boundaries. **(c)** The arbiter compares the reachability probe c^* of a flat plan toward the goal with τ: it may route the episode to the executor alone at the first boundary, and it retires the drafter at the probe's first later crossing, after which the executor targets z_{goal}. **(d)** In each environment τ (dot) is the median latent distance among states that the success criterion treats as equivalent (bar: p10 to p90).

and FF-JEPA (Masip et al., 2026), whose diffusion subgoal model is our drafter, adds a subgoal chain on LeWM. Hierarchical diffusion planners (Janner et al., 2022; Chen et al., 2024) fix subgoal spacing and re-planning schedules in advance. HWM (Zhang et al., 2026) and Hi-LeWM (Caselli et al., 2026) plan hierarchically over latent world models, and Fast-LeWM (Gao & Xu, 2026) speeds up planning over the same LeWM checkpoints. Hi-LeWM's hierarchy, reproduced with its released code at its own protocol, does not improve on flat planning on PushT (Appendix L).

Draft–check–accept decoding (Leviathan et al., 2023; Chen et al., 2023) pairs a cheap drafter with an *exact* acceptance test; our accept rule keeps that structure but replaces the exact test with an offline-calibrated tolerance, so no guarantee carries over. Concurrent SAGE (Cheng et al., 2026a) drafts subgoal latents on LeWM consumed open-loop, and concurrent GC-IDM (Nguyen et al., 2026) amortizes goal-reaching on the same frozen latents into an inverse-dynamics policy with no search and no trust signal; it is both our strongest baseline and an executor inside the layer. Concurrent LeFlow (Huang et al., 2026a) amortizes the same search into a rectified-flow latent path checked by imagined rollout through the frozen predictor, whereas the accept rule checks against the achieved state. Appendix M discusses further concurrent work. Re-solving only when the state leaves a tube around the prediction is *event-triggered control* (Tabuada, 2007; Heemels et al., 2012); AdaReP (Cheng et al., 2026c) applies that trigger to the cached action plan of a world-model MPC with an online-adapted tolerance, RDM (Zhou et al., 2023) regenerates a diffusion plan when its likelihood falls, and DreamLedger (Li et al., 2026b) gates imagination by execution-settled credit. The accept rule instead governs a drafted subgoal sequence, tested against the achieved latent with a tolerance calibrated from the task's success criterion.

3 METHOD

SURVEYOR is a training-free layer that decomposes a distant goal for an unchanged executor. It has three parts: a drafter that proposes latent subgoals, an accept rule that decides at each replanning boundary whether the remaining draft is kept, and an arbiter that decides whether decomposition is needed at all. The two decisions operate on a frozen planning stack and use tolerances calibrated offline by one rule.

3.1 SETTING

An episode starts at an observation o_0 and is given a goal observation o_{goal} recorded t environment steps ahead of it; we call t the *goal distance*, since it is the axis along which flat planning fails. Control proceeds in *replanning intervals* of fixed length, and the time at which one interval ends and the executor is handed its next target is a *replanning boundary*; let r index these boundaries and t_r denote the corresponding environment time. A frozen encoder E maps the current and goal observations to

$$z_r = E(o_{t_r}), \qquad z_{\text{goal}} = E(o_{\text{goal}}). \tag{1}$$

Given a target latent, the *executor* selects actions through online search or an amortized policy and acts for one replanning interval; we treat it as a black box that returns, in addition to its actions, the predicted terminal latent of the plan it selected. Every decision of the layer compares two latents through the normalized discrepancy

$$\text{rel}(a, b) = \frac{\|a - b\|_2}{\|a\|_2}, \tag{2}$$

where the reference latent a is nonzero. Normalizing by its norm makes discrepancies comparable across environments, so that a tolerance τ, calibrated offline as Section 3.5 describes, can serve every comparison in the layer.

3.2 DRAFTER: PROPOSING LATENT SUBGOALS

A drafter $G(\cdot\,; k)$ maps an anchor latent to a block $B = (w_1, \ldots, w_N)$ of N waypoints spaced S environment steps apart, using k network evaluations per draft; additional conditioning inputs are omitted for brevity. Each waypoint is a latent the executor is asked to reach within one replanning interval, so the executor never plans farther than S steps ahead whatever the goal distance. Accept-rule checks are aligned with the spacing S, so that each achieved latent is compared with the waypoint assigned to the same elapsed time; the executor's replanning schedule is fixed, and SURVEYOR adapts only the drafting decisions at these boundaries. The simplest consumption policy, *every-step drafting*, discards the block after one interval and re-drafts from the achieved latent at every boundary; it is the policy of FF-JEPA (Masip et al., 2026) and the reference point for the two decisions below.

3.3 ACCEPT RULE: DECIDING WHEN TO RE-DRAFT

We use execution–plan consistency as an online criterion for continued reuse. At a replanning boundary r, let B_r denote the current draft and $j_r \in \{1, \ldots, N\}$ the index of the waypoint supplied to the executor. After pursuing w_{r,j_r} for one interval, the planner observes z_{r+1} and computes $\delta_{r+1} = \text{rel}(z_{r+1}, w_{r,j_r})$. If drafting remains active, the block and its index are updated as

$$(B_{r+1}, j_{r+1}) = \begin{cases} (B_r, j_r + 1), & \delta_{r+1} \le \tau \text{ and } j_r < N, \\ (G(z_{r+1}; k), 1), & \text{otherwise.} \end{cases} \tag{3}$$

The next target is $w_{r+1,j_{r+1}}$. An episode assigned to the drafter starts with $B_0 = G(z_0; k)$ and $j_0 = 1$. Thus, a successful check permits reuse of the next available waypoint; a failed check or an exhausted block triggers a new draft anchored at the achieved latent. The check uses only the encoded observation and the pursued waypoint, with no learned confidence head or ground-truth state, and authorizes only the next waypoint.

Drafting cost. Let R be the number of replanning decisions and D_R the number of drafter invocations, including initialization and drafts triggered by rejection or block exhaustion. We measure high-level planning cost by the *call ratio* $\rho_R = \frac{D_R}{R}$, which gives $k\rho_R$ drafter evaluations per replanning decision at fixed k, because the drafter is the only component whose cost the consumption policy controls. Every-step drafting has $\rho_R = 1$ and the executor alone $\rho_R = 0$. We count only the boundaries at which drafting is active, so routed episodes and boundaries after retirement, which call no drafter, do not lower the full layer's ratio. Because the call ratio counts drafter calls only, excluding the executor's search and the arbiter's probe, we also report the wall-clock time per episode, which pays both.

3.4 Arbiter: deciding when the layer is not needed

The arbiter determines when the goal should be pursued directly by the executor. At a replanning boundary, a probe planner, by default a flat CEM planner, plans toward z_{goal} with lookahead H, producing a predicted terminal latent $\widehat{z}_{r,H}$. We define the *reachability probe* as the predicted terminal discrepancy

$$c_r^*(H) = \mathrm{rel}(\widehat{z}_{r,H}, z_{\mathrm{goal}}). \tag{4}$$

Whereas δ_{r+1} assesses realized execution, $c_r^*(H)$ is a model-based estimate of goal reachability at lookahead H. The probe planner is the same flat CEM planner for every executor, so that the decision is a property of the goal and the model rather than of the executor; computing the probe with the executor itself is one of the two settings we vary.

Episode-level routing. At the first boundary the probe is computed once, at the executor's own lookahead H, and the episode is routed to the executor alone if

$$c_0^*(H) \le \tau_{\mathrm{r}}; \tag{5}$$

otherwise the drafter is initialized and the executor follows latent waypoints. The routing test is the retirement test of the next paragraph applied at the first boundary.

Replan-level retirement. While the drafter is active, the probe is repeated at subsequent replanning boundaries. Once the probe first satisfies $c_r^* \le \tau_{\mathrm{r}}$, the drafter is retired and the executor targets z_{goal} for the remainder of the episode. The retirement tolerance τ_{r} is the reuse tolerance τ by default; because retirement is irreversible, a stricter $\tau_{\mathrm{r}} < \tau$ is the second arbiter setting we vary. Retirement takes precedence over Equation (3). Disabling the arbiter yields the accept-rule configuration; disabling both yields every-step drafting.

3.5 Calibration

The shared tolerance τ and drafting budget k are selected from offline measurements of the frozen planning stack, completed before closed-loop evaluation and then fixed; Section 4.1 lists the values used, and no closed-loop sweep selected them.

Task-calibrated tolerance. Let $\mathcal{D}_{\mathrm{eq}}$ denote the empirical distribution of $\mathrm{rel}(E(o), E(o'))$ for observation pairs from different episodes in which every body that the encoder observes lies within the success test's tolerance of its counterpart. The pairs match the whole observed scene rather than only the bodies that the success test scores, because frames that agree on a scored object and differ elsewhere are distant in latent space. We set

$$\tau = Q_{\mathcal{D}_{\mathrm{eq}}}(0.5), \tag{6}$$

where $Q_{\mathcal{D}}$ denotes the quantile function, so that τ is the typical latent variation among task-equivalent observations. We apply the same tolerance to the execution discrepancy and to the routing test, and calibrate it in each latent space when the two tests read different spaces. Ground-truth annotations are used only to construct the offline calibration pairs.

Drafting budget. For each candidate budget k, fixed before evaluation, we generate repeated drafts under fixed conditioning and measure their dispersion with rel, and we select the smallest budget at the onset of the dispersion plateau, where further sampler evaluations no longer change it; if no plateau is observed, the rule does not determine k.

4 EXPERIMENTS

We separate the drafter's contribution from that of our two decisions: gains over the *vanilla* executor at long range come mostly from decomposition, which the drafter provides, whereas gains over *every-step drafting*, in drafter calls and in short-range success, come from the accept rule and the arbiter.

4.1 EXPERIMENTAL SETUP

Pre-trained models and executors. Every experiment instantiates the components of Section 3.1 on the frozen JEPA encoder released with LeWM (Maes et al., 2026b). The drafter is FF-JEPA's diffusion subgoal model (Masip et al., 2026), which emits $N{=}3$ subgoal latents spaced $S{=}10$ steps apart, one checkpoint per environment. The *search executors* are LeWM's CEM controller (De Boer et al., 2005), the iCEM (Pinneri et al., 2021) and MPPI (Williams et al., 2017) samplers and an AdamW gradient solver at LeWM's released configuration, all run through the planning interface of stable-worldmodel (Maes et al., 2026a); GC-IDM (Nguyen et al., 2026), trained with the authors' released code, is the amortized executor. Under SURVEYOR every search executor plans ten steps ahead, and a routed episode is handed to that same ten-step planner. An executor run without SURVEYOR is called *vanilla*; for each setting we report the better of its twenty-five-step and its ten-step planner, and on Two-Room only CEM has both (Appendix A).

Environments and protocol. We evaluate on PushT, Reacher, OGBench-Cube and Two-Room. In every environment the goal is the hindsight frame t steps ahead, success is the benchmark's shipped criterion, and the step budget is $2t$. Goal distances run from $t{=}25$, inside the vanilla executor's twenty-five-step lookahead, to $t{=}150$: PushT and Cube use $t \in \{25, 50, 75, 100, 150\}$, Reacher $t \in \{25, 50, 100, 150\}$ and Two-Room $t \in \{25, 50, 75\}$, since its episodes last at most 101 steps. On Cube the block moves at least 8 cm between start and goal, so that no episode is solved at step zero. Every setting is run eight times with different random seeds, which vary the sampling of the solvers and the drafter. PushT, Reacher and Cube are evaluated on fixed, precomputed episode sets, 256 episodes on PushT and 128 on the other two, and on Two-Room each run draws its own 64 episodes. All configurations of an executor are compared on the same episodes. We report the mean success rate over the eight runs; its standard error is at most 3.0 points in any setting (Appendix A). A difference between two means with standard errors s_1 and s_2 is said to exceed two pooled standard errors when it exceeds $2\sqrt{s_1^2 + s_2^2}$.

Calibrated constants. The offline rules of Section 3.5 set the tolerance and the drafting budget per environment. The tolerance, the median latent distance between task-equivalent observations, is $\tau{=}0.233$ on PushT, 0.106 on Reacher, 0.794 on Cube and 0.127 on Two-Room, where each drafted waypoint pairs a LeWM latent with a frozen DINOv2 image embedding and the accept rule checks the DINOv2 part, because LeWM's Two-Room latents place task-equivalent observations almost as far apart as random ones; the probe, a LeWM rollout, is read there against $\tau_{\mathrm{r}}{=}1.323$, measured by the same rule in LeWM space. The budget, the onset of the plateau in draft dispersion, is $k{=}3$ on PushT and Cube; Reacher shows no plateau, and there we use $k{=}8$ for the search executors and $k{=}4$ for GC-IDM; Two-Room uses its sampler's default, $k{=}50$. Appendix B reports the measurements, their bootstrap intervals and the sensitivity of the results to them. SURVEYOR computes the probe with CEM at the executor's lookahead and, except on Two-Room, retires at $\tau_{\mathrm{r}} = \tau$; Appendix I reports the other arbiter settings.

Configurations. For each executor X we compare five configurations, adding one component at a time: vanilla X; every-step drafting over X, which is the FF-JEPA setting; a blind chain, which keeps every draft for its N waypoints without checking; the accept rule; and the full layer, X + SURVEYOR. Table 1 compares the first and the last over all goal distances, Table 2 separates the short distances from the long ones, and Table 16 in Appendix F reports the three in between.

4.2 PERFORMANCE ACROSS GOAL DISTANCES

All goal distances. ❶ SURVEYOR raises mean success for eighteen of the twenty executor–environment pairs (Table 1), every pair on PushT and on Cube among them. Over GC-IDM

Table 1: **Success and cost of each executor X, vanilla and under the full layer (X + SURVEYOR).** Success rate (%) is the mean over eight repeated runs and over all goal distances of an environment: t=25, 50, 75, 100 and 150 on PushT and Cube, 25, 50, 100 and 150 on Reacher, and 25, 50 and 75 on Two-Room. Seconds per episode are wall-clock at the longest distance (t=150; Two-Room 75) on one NVIDIA A100, the vanilla executor timed at the lookahead whose success is printed; the call ratio is the number of drafter calls per replanning step while drafting at PushT t=150, one under every-step drafting. In each pair of rows the better value (higher success, fewer seconds) is **bold** on green, and where the two are equal the SURVEYOR value is on amber.

Executor	mean success rate (%)				seconds per episode				call ratio
	PushT	Reacher	Cube	Two-Room	PushT	Reacher	Cube	Two-Room	PushT
CEM (De Boer et al., 2005)	36.8	85.0	39.6	60.4	14.9	4.1	**9.5**	**7.0**	0
CEM + SURVEYOR	**95.6**	**95.2**	**45.8**	**69.6**	**12.1**	**3.2**	12.8	9.1	0.58
GC-IDM (Nguyen et al., 2026)	88.7	**98.4**	49.3	100.0	2.1	**1.6**	**2.6**	**0.7**	0
GC-IDM + SURVEYOR	**96.1**	94.8	**57.8**	100.0	2.1	1.8	2.9	2.5	0.73
iCEM (Pinneri et al., 2021)	35.6	87.7	40.6	60.6	16.0	3.7	**8.5**	**7.0**	0
iCEM + SURVEYOR	**92.3**	**96.2**	**42.7**	**81.4**	**13.1**	**2.9**	13.0	8.0	0.64
MPPI (Williams et al., 2017)	36.1	84.0	25.4	43.0	15.8	4.6	**12.9**	**8.1**	0
MPPI + SURVEYOR	**94.1**	**90.9**	**27.2**	**43.5**	**12.4**	**3.6**	14.1	11.4	0.67
Gradient (Maes et al., 2026a)	47.2	93.3	41.1	64.1	50.4	**9.5**	**23.3**	25.6	0
Gradient + SURVEYOR	**96.4**	**96.2**	**42.7**	**71.6**	**46.3**	10.3	37.0	25.6	0.60

Table 2: **Short- and long-range success of each executor X, vanilla and under the full layer.** Success rate (%), the mean over eight repeated runs and over the short goal distances (t=25 and 50) or the long ones (t=75, 100 and 150; Reacher 100 and 150; Two-Room 75). In each pair of rows the higher value is **bold** on green, and where the two are equal the SURVEYOR value is on amber.

Executor	PushT		Reacher		Cube		Two-Room	
	short	long	short	long	short	long	short	long
CEM	73.5	12.3	94.5	75.5	49.6	33.0	69.5	42.2
CEM + SURVEYOR	**96.2**	**95.2**	**97.9**	**92.5**	**55.9**	**39.0**	**72.4**	**64.1**
GC-IDM	**98.7**	82.0	**99.6**	**97.3**	58.1	43.5	100.0	100.0
GC-IDM + SURVEYOR	98.0	**94.9**	94.4	95.2	**64.7**	**53.2**	100.0	100.0
iCEM	72.3	11.1	97.1	78.3	50.9	33.7	70.7	40.4
iCEM + SURVEYOR	**93.5**	**91.5**	**98.5**	**93.9**	**53.2**	**35.7**	**84.0**	**76.4**
MPPI	71.4	12.6	85.7	82.2	33.5	20.1	**48.0**	33.0
MPPI + SURVEYOR	**94.6**	**93.8**	**87.4**	**94.5**	**34.6**	**22.2**	47.5	**35.5**
Gradient	85.3	21.8	95.7	91.0	48.4	36.3	74.2	43.9
Gradient + SURVEYOR	**97.0**	**95.9**	**97.7**	**94.8**	**50.4**	**37.5**	**77.4**	**60.0**

the gains are smaller because GC-IDM, which maps the current and goal latents to actions without searching the world model, degrades little with distance: vanilla, it solves 98.4% of Reacher episodes and every Two-Room episode, which leaves the layer little to add. Where GC-IDM does fail, the layer lifts it, from 62.7% to 94.2% at PushT t=150 and by 8.5 points on Cube, the largest Cube gain of any executor. The one pair that falls, GC-IDM on Reacher (3.6 points), is explained below. ❷ No executor under SURVEYOR falls below 81.0% on any PushT or Reacher setting, whereas every vanilla search executor falls below 10% on at least one, and vanilla GC-IDM falls to 62.7%. ❸ SURVEYOR makes fewer drafter calls than every-step drafting on PushT, Reacher and Cube: at the longest distance its call ratio, the number of drafter calls per replanning step while drafting, is 0.58 to 0.73 on PushT, 0.78 to 0.99 on Reacher, 0.48 to 0.56 on Cube and 0.86 to 1.00 on Two-Room, against one for every-step drafting, and routed episodes make none (Section 4.3).

Short horizon ($t \leq 50$). Decomposition by itself can cost success at short range: at Reacher t=25 every-step drafting lowers every executor, CEM from 95.7% to 65.2% and GC-IDM from 100% to

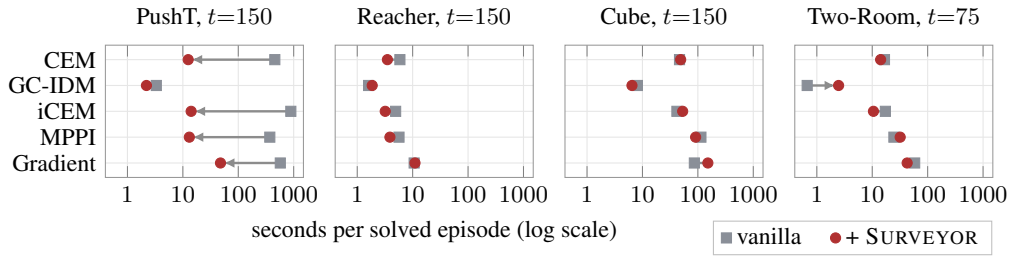

Figure 3: **Cost per solved episode.** Wall-clock seconds per episode divided by the success rate, at the longest goal distance of each environment on one NVIDIA A100, means over eight repeated runs, with the vanilla executor timed at the lookahead whose success Table 1 prints. The layer is cheaper where the circle lies to the left of the square; Appendix H gives the values.

80.6%, because a chain of subgoals can lead an executor away from a goal it could have reached directly. The arbiter hands such episodes back to the executor (Section 4.4), and ❹ on PushT, Reacher and Cube SURVEYOR raises the short-range success of every search executor (Table 2), beyond two pooled standard errors in ten of these twelve pairs; the largest gains are at PushT t=50, where the vanilla search executors solve 47.0% to 72.4% of episodes. On Two-Room the arbiter routes 72% of the t=25 episodes to the executor (Table 17), where every-step drafting would lower CEM from 78.3% to 67.4%, and short-range success rises over CEM, iCEM and the gradient solver and stays within a point over MPPI. On Reacher, where vanilla GC-IDM is near 100%, the CEM probe hands the drafter episodes that GC-IDM would have solved directly, and SURVEYOR trails it by 2 to 6 points at every distance.

Long horizon ($t \geq 75$). ❺ Over the long distances eighteen pairs rise, seventeen beyond two pooled standard errors (Table 2), and on PushT and Reacher the gap widens with distance (Figure 1): at PushT t=150 the vanilla search executors solve 1.8% to 8.8% of episodes and SURVEYOR 93.1% to 97.1%, and over GC-IDM success rises from 62.7% to 94.2%. Most of this lift comes from decomposition itself. Every-step drafting, with neither of our decisions, already solves 92.1% to 96.9% at PushT t=150 and 91.7% to 93.3% at Reacher t=150, where the vanilla search executors solve 69.6% to 89.3% (Appendix A); SURVEYOR stays within two points of that success with fewer drafter calls (Section 4.3). Because the layer turns one long problem into several short ones, its lift tracks the executor's success at short range. The search executors solve only 42% to 61% of Cube episodes even at t=25, and the lift over them is 1.2 to 6.0 points; over GC-IDM, the strongest Cube executor at short range, it is 9.7. Cutting PushT's CEM to that short-range level leaves every-step drafting under 1% at t=150 (Appendix D). On Two-Room SURVEYOR gains 2.5 to 36.0 points at t=75 over the four search executors.

4.3 PLANNING EFFICIENCY AND DRAFTER COST

We next change only how drafts are consumed (Table 16). We observe that ❶ the accept rule keeps the success of every-step drafting with fewer drafter calls. Over the four search executors, mean success over PushT and Reacher under the accept rule is within 0.5 points of every-step drafting. At PushT t=150 the accept rule is ahead of every-step drafting for three of them and within half a point for the fourth, while making 0.58 to 0.67 as many drafter calls. On Reacher at t=25 and t=50 the accept rule alone loses ground to every-step drafting (at t=50 over CEM, 88.3% to 82.6%), which the arbiter recovers (Section 4.4). ❷ On Reacher the check itself matters, not only the reuse. A blind chain, which reuses every draft without checking, makes about a third as many drafter calls as every-step drafting but trails the accept rule by 3 to 18 points on Reacher for every search executor. On PushT it also trails the accept rule over MPPI, by 4 to 8 points beyond t=25, and on Cube it matches or beats the accept rule (Appendix A).

End-to-end cost. ❸ Per solved episode at the longest distance, SURVEYOR is the cheaper of the two in thirteen of the twenty executor–environment pairs (Figure 3). Its time per episode is lower than the vanilla executor's for every search executor on PushT and every sampling solver on Reacher, where solved episodes end early, and higher or equal elsewhere (Table 1). The added time is the

arbiter's probe, a flat CEM plan at each boundary: over CEM on PushT it raises the time per episode from the accept rule's 8.1 s to 12.1 s, and it buys the short-range success of Section 4.4. The drafter takes below 1.5% of the time spent on drafting and search (Appendix H), so with this drafter the accept rule saves drafter calls rather than time.

4.4 Short-range recovery by the arbiter

We then add the arbiter, which gives SURVEYOR, and report it in Tables 1 and 16; at short range it mostly routes episodes to the executor, and at long range it retires the drafter partway through most Reacher, Cube and Two-Room episodes (Table 17). We observe that ❶ SURVEYOR recovers most of the short-range success that decomposition had lost. At Reacher $t{=}25$ the arbiter routes most episodes to the executor, and success returns to 95.3%–98.8% over CEM, iCEM, the gradient solver and GC-IDM, against 53.0%–63.6% for the accept rule alone. Over MPPI, whose flat planner solves 81.0% of those episodes, SURVEYOR reaches that same 81.0%. ❷ Routing does more than fall back on the flat planner. SURVEYOR routes 91% of these episodes to the ten-step flat planner, yet it solves more of them than that planner alone: 98.2% against 95.7% over CEM and 98.8% against 97.4% over iCEM, both beyond two pooled standard errors. The probe routes to the flat planner the episodes that planner can finish and decomposes the rest. Over MPPI and the gradient solver the two are level. ❸ The worst PushT or Reacher setting rises from 52%–64% for the accept rule to 81.0%–93.5% on every executor, and the mean to 92.7%–96.3%, at the accept rule's call ratio at PushT $t{=}150$.

What the arbiter costs. The arbiter loses success when it routes or retires too early and hands the episode to an executor that cannot finish it: at PushT $t{=}50$ it costs 3.1 points over CEM and 4.5 over iCEM, at Cube $t{=}25$ over iCEM 4.7, and at Cube $t{=}50$ and $t{=}75$ over GC-IDM 17.6 and 13.3. There it routes 57% and 40% of episodes to GC-IDM alone, weaker than GC-IDM with the drafter. A stricter retirement tolerance removes the PushT loss over CEM and moves no Reacher setting by over 1.4 points, and with GC-IDM's own probe it raises these two Cube settings to 71.4% and 64.3% (Appendix I).

5 Discussion and limitations

What each component contributes. The drafter extends the range: long-range success is already high under every-step drafting. The accept rule cuts drafter calls at the long-range success of every-step drafting on PushT and Reacher, and the arbiter restores success at short goal distances, where subgoals hurt. Neither decision uses the executor's internal state: the accept rule compares the achieved latent with the subgoal and the arbiter a plan computed from the world model, so it routes the same fraction of episodes for every search executor to within one point (Appendix G).

Limitations. All results are in simulation, with one frozen world model per environment. Over the flow planner LeFlow the layer lowers success, and LeFlow itself loses 11.5 to 56.8 points at $t{=}25$ when its horizon is cut to the ten steps between subgoals (Appendix K).

6 Conclusion

We introduced SURVEYOR, a training-free decomposition layer that proposes latent subgoals for an unchanged executor, reuses drafted plans while they agree with realized execution, and uses predicted goal reachability to decide when decomposition is not needed. Over five executors and four environments the layer lifts PushT success at the longest distance from single digits to above 92% on every search executor, at 0.58 to 0.67 of the drafter calls of every-step drafting, and recovers most of the short-range success that decomposition gives up. These findings support treating high-level plans as reusable objects whose continued use is guided by execution feedback and whose use at all is guided by predicted reachability.

Reproducibility statement

Every experiment behind Table 1 was pre-registered before it was run, with its predictions, falsifiers and decision rules written down in advance. The anonymized supplementary material contains the code of the layer and of the executor interfaces, the exact command of every run, the fixed evaluation episodes, the calibration measurements of Section 3.5, the per-run results, and the scripts that regenerate every table and figure of this paper from those results. The datasets, the LeWM world models and the GC-IDM code are public, and the solvers are those of a public planning library.

AI use statement

Large language models were used solely to refine the wording of selected sentences and to check formatting during manuscript preparation. All ideas, methods, experiments, analyses, and conclusions are the authors' own, and the authors take full responsibility for the content of this paper.

References

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems*, 38:33383–33407, 2026.

Niccolò Caselli, Francesco Massafra, Samuele Punzo, Salvatore Lo Sardo, Ippokratis Pantelidis, and Sathya Kamesh Bhethanabhotla. Mind the gap: Promises and pitfalls of hierarchical planning in LeWorldModel. *arXiv preprint arXiv:2607.12547*, 2026.

Chang Chen, Fei Deng, Kenji Kawaguchi, Caglar Gulcehre, and Sungjin Ahn. Simple hierarchical planning with diffusion. In *International Conference on Learning Representations*, volume 2024, pp. 15979–16002, 2024.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Letian Cheng, Qi Zhang, and Yisen Wang. SAGE: Subgoal-conditioned action generation for latent world model planning. *arXiv preprint arXiv:2607.17973*, 2026a.

Xin Cheng, Xingkai Yu, Chenze Shao, Jiashi Li, Yunfan Xiong, Yi Qian, Jiaqi Zhu, Shirong Ma, Xiaokang Zhang, Jiasheng Ye, et al. DSpark: Confidence-scheduled speculative decoding with semi-autoregressive generation. *arXiv preprint arXiv:2607.05147*, 2026b.

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. *arXiv preprint arXiv:2606.23079*, 2026c.

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.

Tianqi Du, Qi Zhang, Yifei Wang, and Yisen Wang. Beyond the next step: Variable-length latent world models for long-horizon planning. *arXiv preprint arXiv:2606.21775*, 2026.

Baoqi Gao, Ruize Han, Miao Wang, and Song Wang. IMWM: Intuition models complement world models for latent planning. *arXiv preprint arXiv:2606.01626*, 2026.

Yuntian Gao and Xiangyu Xu. Fast LeWorldModel. *arXiv preprint arXiv:2606.26217*, 2026.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in Neural Information Processing Systems*, 31, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nick Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Wilhelmus PMH Heemels, Karl Henrik Johansson, and Paulo Tabuada. An introduction to event-triggered and self-triggered control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 3270–3285. IEEE, 2012.

Jiaming Hu, Yan Zheng, and Tian Wang. SCALE: State-calibrated latent embeddings for JEPA planning in the right geometry. *arXiv preprint arXiv:2608.16287*, 2026.

Hsiang-Wei Huang, Jianxu Shangguan, Junbin Lu, and Jenq-Neng Hwang. LeFlow: Generative latent flow planning for world models, 2026a. URL https://arxiv.org/abs/2608.24855.

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. *arXiv preprint arXiv:2608.14125*, 2026b.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Liangyu Li, Shengzhi Wang, and Qingwen Liu. Beyond Euclidean proximity: Repairing latent world models with horizon-matched trajectory reachability metrics. *arXiv preprint arXiv:2605.22164*, 2026a.

Xianyao Li, Ruitong Tian, Rui Min, Fang Xu, and Eric Jing Du. DreamLedger: Where to refuse world-model imagination using execution-settled credit. *arXiv preprint arXiv:2608.23863*, 2026b.

Yuejiang Liu, Jubayer Hamid, Annie Xie, Yoonho Lee, Max Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations*, volume 2025, pp. 4594–4627, 2025.

Zihan Liu, Yuzhe Zhuang, Yuanzu Li, Wanshuang Gou, Jiahong Liu, Min Zhou, and Menglin Yang. ProWorld: Progress-aware hyperbolic world models for long-horizon visual goal reaching. *arXiv preprint arXiv:2608.01926*, 2026.

Lucas Maes, Quentin Le Lidec, Luiz Facury, Nassim Massaudi, Ayush Chaurasia, Francesco Capuano, Richard Gao, Taj Gillin, Dan Haramati, Damien Scieur, Yann LeCun, and Randall Balestriero. stable-worldmodel: A platform for reproducible world modeling research and evaluation. *arXiv preprint arXiv:2605.21800*, 2026a.

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*, 2026b.

Sergi Masip, Jonathan Swinnen, Yutong Hu, Renaud Detry, and Tinne Tuytelaars. FF-JEPA: Long-horizon planning in world models with latent planners. *arXiv preprint arXiv:2606.09311*, 2026.

Hoang Nguyen, Xiaohao Xu, and Xiaonan Huang. Latent geometry beyond search: Amortizing planning in world models. *arXiv preprint arXiv:2605.08732*, 2026.

Phu Pham and Aniket Bera. Latent energy action planning with world models. *arXiv preprint arXiv:2609.03294*, 2026.

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.

Joyjeet Singh. The evaluation protocol determines the result: An independent reproduction of LeWorldModel on TwoRoom. *arXiv preprint arXiv:2608.10145*, 2026.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim GJ Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. *Advances in Neural Information Processing Systems*, 38:43905–43941, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, volume 2024, pp. 15078–15097, 2024.

Junhan Sun, Hao Zhao, and Guofeng Zhang. INTACT: Isomorphic intent-to-action learning for search-free world models. *arXiv preprint arXiv:2607.26056*, 2026.

Paulo Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

Kailin Wang, Haoxiang Jie, Yaoyuan Yan, Jiacheng Zhou, and Zhiyou Heng. EV-WM: Event-verified world models for long-horizon robotic manipulation. *arXiv preprint arXiv:2606.13053*, 2026.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic MPC for model-based reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1714–1721. IEEE, 2017.

Pengyuan Wu, Pingrui Zhang, Zhigang Wang, Dong Wang, Bin Zhao, and Xuelong Li. Closed-loop action chunks with dynamic corrections for training-free diffusion policy. *arXiv preprint arXiv:2603.01953*, 2026.

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. *arXiv preprint arXiv:2604.03208*, 2026.

Zhengshu Zhang and Zhiyuan Li. ARC-Bench: Closed-loop replanning masks broken action ranking in frozen JEPA world models. *arXiv preprint arXiv:2609.05461*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

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*, 2023. arXiv:2310.09629.

A FULL RESULTS WITH STANDARD ERRORS

Tables 3 to 6 report every configuration at every goal distance, as the mean over eight evaluation seeds with its standard error. The "vanilla" row is the executor's better flat planner per cell, of the twenty-five-step and the ten-step horizon; on Two-Room only CEM has both. Indented rows are the arbiter variants and controls of Appendix I, and "blind chain (2 waypoints)" keeps each draft for two waypoints instead of three. GC-IDM has no sampler of its own, so its seeds differ little and coincide on Reacher.

Table 3: **All configurations on PushT.** Success rate (%) by goal distance t, with the tolerance at its measured value τ=0.233. Mean over eight evaluation seeds with its standard error. Rows of the full layer are in **bold**; indented rows are its variants and controls (Appendix I).

Executor	Configuration	$t = 25$	$t = 50$	$t = 75$	$t = 100$	$t = 150$
CEM (LeWM)	vanilla	96.4 ±0.3	50.6 ±0.4	19.9 ±0.3	13.7 ±0.5	3.3 ±0.2
	drafter, every-step (S=10)	99.2 ±0.2	95.3 ±0.4	94.8 ±0.4	94.0 ±0.5	95.8 ±0.6
	drafter, every-step (S=25)	95.1 ±0.2	91.2 ±0.4	89.8 ±0.7	91.7 ±0.4	92.8 ±0.7
	blind chain (2 waypoints)	98.7 ±0.1	96.9 ±0.4	95.3 ±0.5	95.3 ±0.3	97.2 ±0.3
	blind chain	98.5 ±0.1	95.6 ±0.4	94.8 ±0.4	94.9 ±0.5	94.9 ±0.5
	accept rule	98.8 ±0.2	96.2 ±0.3	96.1 ±0.4	95.3 ±0.2	97.7 ±0.4
	full (accept rule + arbiter)	**99.4 ±0.1**	**93.1 ±0.8**	**94.9 ±0.5**	**93.8 ±0.5**	**97.1 ±0.5**
	τ_r=0.10	99.1 ±0.1	96.1 ±0.4	96.6 ±0.2	95.1 ±0.4	97.2 ±0.3
	every-step under the arbiter	99.6 ±0.2	91.8 ±0.5	93.0 ±0.3	92.2 ±0.6	96.3 ±0.3
	plan-free router	99.0 ±0.2	96.7 ±0.3	95.9 ±0.3	95.2 ±0.3	97.8 ±0.3
GC-IDM	vanilla	100.0 ±0.0	97.3 ±0.1	92.2 ±0.2	91.1 ±0.2	62.7 ±0.2
	drafter, every-step	99.3 ±0.1	96.2 ±0.3	94.2 ±0.3	94.9 ±0.3	94.9 ±0.3
	blind chain	99.2 ±0.1	94.9 ±0.4	93.7 ±0.3	95.2 ±0.3	95.7 ±0.3
	accept rule	99.4 ±0.1	95.5 ±0.4	95.7 ±0.3	94.9 ±0.3	94.0 ±0.3
	full (accept rule + arbiter)	**99.8 ±0.1**	**96.1 ±0.4**	**94.9 ±0.3**	**95.5 ±0.3**	**94.2 ±0.3**
	executor as probe	99.6 ±0.1	96.6 ±0.5	94.7 ±0.5	95.7 ±0.3	93.7 ±0.5
	τ_r=0.10	99.4 ±0.1	96.0 ±0.5	95.8 ±0.3	95.0 ±0.3	93.9 ±0.5
	executor as probe, τ_r=0.10	99.4 ±0.1	95.8 ±0.3	95.0 ±0.5	95.3 ±0.4	93.6 ±0.3
iCEM	vanilla	97.6 ±0.2	47.0 ±0.5	19.5 ±0.4	11.9 ±0.3	1.8 ±0.1
	drafter, every-step	99.0 ±0.1	92.3 ±0.5	89.4 ±0.4	88.0 ±0.4	92.1 ±0.4
	blind chain (2 waypoints)	99.1 ±0.1	95.8 ±0.5	95.4 ±0.5	94.4 ±0.4	95.0 ±0.4
	blind chain	99.0 ±0.1	95.3 ±0.5	95.4 ±0.6	93.7 ±0.5	92.6 ±0.6
	accept rule	99.0 ±0.1	94.2 ±0.2	93.4 ±0.7	92.6 ±0.5	93.9 ±0.4
	full (accept rule + arbiter)	**97.4 ±0.2**	**89.7 ±0.8**	**91.0 ±0.5**	**90.4 ±0.4**	**93.1 ±0.4**
	executor as probe	99.4 ±0.1	94.2 ±0.5	93.7 ±0.5	92.8 ±0.5	93.7 ±0.5
	τ_r=0.10	99.3 ±0.1	94.0 ±0.4	93.3 ±0.5	92.3 ±0.4	93.8 ±0.4
	executor as probe, τ_r=0.10	99.3 ±0.1	94.4 ±0.5	92.8 ±0.3	92.4 ±0.4	94.0 ±0.4
	every-step under the arbiter	97.8 ±0.2	86.1 ±0.9	86.7 ±1.1	87.2 ±0.7	90.7 ±0.3
	plan-free router	98.9 ±0.1	94.7 ±0.3	92.9 ±0.5	92.1 ±0.3	94.2 ±0.5
MPPI	vanilla	91.5 ±0.6	51.3 ±1.2	21.1 ±0.6	12.5 ±0.7	4.2 ±0.4
	drafter, every-step	97.6 ±0.2	93.5 ±0.7	93.1 ±0.5	94.3 ±0.2	94.8 ±0.3
	blind chain (2 waypoints)	96.8 ±0.5	93.0 ±0.6	91.8 ±0.7	93.2 ±0.7	93.6 ±0.3
	blind chain	96.4 ±0.4	89.5 ±0.8	88.1 ±0.4	89.2 ±0.5	87.1 ±0.8
	accept rule	97.7 ±0.6	93.8 ±0.6	94.4 ±0.5	95.1 ±0.4	94.9 ±0.5
	full (accept rule + arbiter)	**97.1 ±0.4**	**92.1 ±0.5**	**92.4 ±0.6**	**94.5 ±0.7**	**94.4 ±0.4**
	executor as probe	97.1 ±0.4	93.4 ±0.6	93.2 ±0.3	95.5 ±0.5	94.7 ±0.3
	τ_r=0.10	97.6 ±0.3	94.7 ±0.6	95.0 ±0.4	95.4 ±0.4	95.4 ±0.5
	executor as probe, τ_r=0.10	97.9 ±0.4	93.7 ±0.3	94.2 ±0.6	95.7 ±0.3	95.3 ±0.3
Gradient (AdamW)	vanilla	98.1 ±0.3	72.4 ±0.7	34.7 ±0.9	21.9 ±0.6	8.8 ±0.5
	drafter, every-step	98.8 ±0.2	95.9 ±0.4	94.2 ±0.4	94.8 ±0.3	96.9 ±0.5
	blind chain (2 waypoints)	98.5 ±0.1	96.6 ±0.5	95.4 ±0.2	95.6 ±0.5	98.0 ±0.5
	blind chain	98.5 ±0.3	95.6 ±0.4	94.9 ±0.5	94.9 ±0.5	95.7 ±0.4
	accept rule	98.7 ±0.1	96.2 ±0.2	95.9 ±0.3	95.8 ±0.3	96.8 ±0.3
	full (accept rule + arbiter)	**99.2 ±0.2**	**94.9 ±0.2**	**95.6 ±0.4**	**95.4 ±0.3**	**96.9 ±0.4**
	executor as probe	98.8 ±0.2	96.0 ±0.2	96.0 ±0.4	95.8 ±0.3	97.3 ±0.3
	τ_r=0.10	99.0 ±0.1	96.5 ±0.4	95.8 ±0.4	96.1 ±0.3	97.0 ±0.3
	executor as probe, τ_r=0.10	99.1 ±0.2	96.2 ±0.4	96.4 ±0.4	96.1 ±0.3	97.5 ±0.2

Table 4: **All configurations on Reacher.** Success rate (%) by goal distance t, with the tolerance at its measured value τ=0.106. Mean over eight evaluation seeds with its standard error. Rows of the full layer are in **bold**; indented rows are its variants and controls (Appendix I).

Executor	Configuration	$t = 25$	$t = 50$	$t = 100$	$t = 150$
CEM (LeWM)	vanilla	95.7 ±0.3	93.3 ±0.4	81.3 ±0.4	69.6 ±0.5
	drafter, every-step (S=10)	65.2 ±1.6	88.3 ±0.9	95.2 ±0.6	92.7 ±0.7
	drafter, every-step (S=25)	51.9 ±1.3	79.6 ±1.8	89.5 ±0.9	89.7 ±1.1
	blind chain (2 waypoints)	57.7 ±0.7	78.7 ±0.9	91.1 ±0.9	91.4 ±0.5
	blind chain	50.5 ±1.8	67.5 ±1.2	82.1 ±0.9	86.2 ±0.4
	accept rule	62.8 ±1.4	82.6 ±0.8	94.5 ±0.6	93.1 ±0.5
	full (accept rule + arbiter)	**98.2 ±0.4**	**97.6 ±0.3**	**94.0 ±0.5**	**90.9 ±0.6**
	τ_r=0.10	98.5 ±0.4	98.8 ±0.3	95.4 ±0.7	90.1 ±0.6
	every-step under the arbiter	99.0 ±0.3	97.8 ±0.4	95.0 ±0.6	89.5 ±0.8
	plan-free router	63.0 ±1.4	82.5 ±0.8	94.2 ±0.7	93.0 ±0.5
GC-IDM	vanilla	100.0 ±0.0	99.2 ±0.0	96.9 ±0.0	97.7 ±0.0
	drafter, every-step	80.6 ±1.5	96.1 ±0.4	96.5 ±0.3	95.3 ±0.4
	blind chain	54.1 ±1.3	71.9 ±1.9	90.3 ±0.5	93.2 ±0.5
	accept rule	60.7 ±1.2	82.3 ±1.3	95.5 ±0.6	94.6 ±0.5
	full (accept rule + arbiter)	**95.3 ±0.4**	**93.5 ±0.6**	**95.0 ±0.4**	**95.3 ±0.5**
	executor as probe	64.3 ±1.8	85.8 ±0.8	96.0 ±0.4	95.2 ±0.5
	τ_r=0.10	94.0 ±0.3	93.0 ±0.7	96.3 ±0.4	95.1 ±0.5
	executor as probe, τ_r=0.10	62.0 ±1.3	85.1 ±1.2	96.2 ±0.5	94.8 ±0.6
iCEM	vanilla	97.4 ±0.1	96.8 ±0.2	82.4 ±0.2	74.1 ±0.4
	drafter, every-step	61.5 ±2.8	83.8 ±0.6	93.3 ±0.4	93.1 ±0.6
	blind chain (2 waypoints)	49.8 ±1.8	73.4 ±0.5	89.9 ±0.6	91.7 ±0.6
	blind chain	42.8 ±1.9	61.9 ±1.2	81.5 ±1.1	86.9 ±0.7
	accept rule	53.0 ±1.4	79.7 ±1.7	90.3 ±0.4	93.1 ±0.5
	full (accept rule + arbiter)	**98.8 ±0.3**	**98.2 ±0.2**	**95.9 ±0.4**	**91.9 ±0.6**
	executor as probe	97.8 ±0.3	98.2 ±0.2	96.1 ±0.3	94.1 ±0.2
	τ_r=0.10	99.0 ±0.3	98.5 ±0.3	96.4 ±0.4	93.9 ±0.5
	executor as probe, τ_r=0.10	97.3 ±0.4	97.8 ±0.7	96.4 ±0.4	93.5 ±0.7
	every-step under the arbiter	99.0 ±0.3	98.3 ±0.3	96.2 ±0.5	93.5 ±0.8
	plan-free router	52.5 ±1.2	80.0 ±1.8	91.9 ±0.8	93.2 ±0.7
MPPI	vanilla	81.0 ±0.8	90.4 ±0.7	83.0 ±0.7	81.5 ±0.8
	drafter, every-step	54.2 ±1.2	76.0 ±1.4	93.0 ±0.5	91.7 ±0.4
	blind chain (2 waypoints)	48.1 ±2.0	68.7 ±1.1	89.4 ±0.9	91.0 ±0.8
	blind chain	47.8 ±2.0	64.4 ±0.8	83.6 ±0.8	89.2 ±0.5
	accept rule	52.4 ±1.3	78.5 ±1.2	92.7 ±0.6	92.5 ±0.6
	full (accept rule + arbiter)	**81.0 ±1.1**	**93.8 ±1.1**	**96.2 ±0.7**	**92.8 ±0.8**
	executor as probe	82.6 ±0.6	93.8 ±0.5	94.4 ±0.7	92.1 ±0.5
	τ_r=0.10	79.1 ±1.4	94.7 ±0.6	97.0 ±0.2	93.8 ±0.5
	executor as probe, τ_r=0.10	80.1 ±1.0	94.8 ±0.6	95.4 ±0.7	92.9 ±0.5
Gradient (AdamW)	vanilla	97.2 ±0.7	94.1 ±0.4	92.8 ±0.6	89.3 ±0.7
	drafter, every-step	66.8 ±1.3	88.1 ±0.6	95.3 ±0.6	93.3 ±0.8
	blind chain (2 waypoints)	59.7 ±2.4	80.0 ±0.9	93.2 ±0.5	92.9 ±0.2
	blind chain	52.2 ±1.4	71.0 ±1.0	87.8 ±0.5	89.7 ±0.5
	accept rule	63.6 ±1.1	86.6 ±1.2	95.5 ±0.3	93.5 ±0.6
	full (accept rule + arbiter)	**96.5 ±0.5**	**98.8 ±0.1**	**96.4 ±0.3**	**93.3 ±1.0**
	executor as probe	94.1 ±0.6	98.5 ±0.3	96.0 ±0.2	94.2 ±0.7
	τ_r=0.10	96.1 ±0.6	99.0 ±0.3	97.4 ±0.5	92.6 ±0.7
	executor as probe, τ_r=0.10	93.3 ±0.8	98.8 ±0.3	96.0 ±0.3	93.4 ±0.5

Table 5: **All configurations on Cube.** Success rate (%) by goal distance t, with the tolerance at its measured value τ=0.794, on fixed populations in which the block moves at least 8 cm. Mean over eight evaluation seeds with its standard error. Rows of the full layer are in **bold**; indented rows are its variants and controls (Appendix I).

Executor	Configuration	$t = 25$	$t = 50$	$t = 75$	$t = 100$	$t = 150$
CEM (LeWM)	vanilla	60.7 ±0.4	38.4 ±0.7	43.8 ±0.5	35.1 ±0.4	20.3 ±0.7
	drafter, every-step (S=10)	59.2 ±0.8	43.4 ±0.7	33.6 ±1.0	36.6 ±0.9	26.0 ±1.6
	blind chain	69.4 ±1.0	50.2 ±0.8	38.1 ±1.0	38.2 ±0.9	32.4 ±1.2
	accept rule	66.0 ±1.8	46.9 ±0.9	34.6 ±1.1	36.4 ±0.8	26.8 ±1.0
	full (accept rule + arbiter)	**66.4 ±0.6**	**45.4 ±1.1**	**49.7 ±0.8**	**41.3 ±0.7**	**26.1 ±0.8**
	τ_r=0.10	66.6 ±1.1	49.9 ±0.7	48.4 ±0.9	40.6 ±0.9	26.7 ±1.5
GC-IDM	vanilla	71.5 ±0.4	44.6 ±0.3	49.0 ±0.3	49.9 ±0.3	31.6 ±0.7
	drafter, every-step	74.9 ±0.9	66.1 ±1.1	62.6 ±0.7	71.2 ±0.8	51.2 ±0.9
	blind chain	80.7 ±0.9	71.2 ±0.6	65.0 ±0.3	70.1 ±0.9	51.1 ±1.0
	accept rule	81.4 ±0.3	70.3 ±0.9	64.2 ±0.7	71.7 ±0.6	49.5 ±1.1
	full (accept rule + arbiter)	**76.7 ±0.6**	**52.7 ±0.5**	**50.9 ±0.6**	**63.1 ±0.8**	**45.5 ±0.8**
	executor as probe	75.2 ±0.2	48.0 ±0.5	51.0 ±0.1	53.2 ±0.3	33.1 ±0.3
	τ_r=0.10	80.9 ±0.7	65.4 ±0.9	52.5 ±0.5	66.9 ±0.5	50.3 ±1.3
	executor as probe, τ_r=0.10	81.0 ±0.7	71.4 ±1.3	64.3 ±0.5	71.5 ±0.8	49.8 ±0.8
iCEM	vanilla	61.3 ±0.4	40.4 ±0.6	46.5 ±0.3	34.3 ±0.5	20.4 ±0.5
	drafter, every-step	65.8 ±0.9	46.5 ±0.8	46.3 ±0.8	35.8 ±0.7	28.4 ±0.7
	blind chain	68.0 ±0.7	48.8 ±0.7	43.2 ±1.0	33.8 ±0.7	31.4 ±0.6
	accept rule	67.2 ±0.6	47.9 ±0.6	41.0 ±1.4	34.5 ±1.1	28.6 ±0.5
	full (accept rule + arbiter)	**62.5 ±0.5**	**43.9 ±0.8**	**50.2 ±0.5**	**32.5 ±0.2**	**24.4 ±0.8**
	executor as probe	66.8 ±0.5	46.2 ±0.8	51.1 ±0.8	37.5 ±0.8	26.2 ±0.8
	τ_r=0.10	65.6 ±0.6	49.1 ±0.5	49.5 ±0.4	35.9 ±0.6	27.7 ±0.7
	executor as probe, τ_r=0.10	72.8 ±0.7	47.7 ±0.6	44.1 ±0.8	33.8 ±0.8	27.9 ±0.9
MPPI	vanilla	41.8 ±1.4	25.3 ±0.7	28.1 ±1.0	20.7 ±0.7	11.3 ±0.6
	drafter, every-step	30.4 ±1.1	25.7 ±0.8	16.4 ±1.1	18.5 ±1.0	11.7 ±1.0
	blind chain	32.4 ±1.1	28.1 ±1.1	16.0 ±1.0	20.6 ±1.2	14.1 ±0.8
	accept rule	32.3 ±0.8	28.4 ±1.2	17.0 ±0.7	20.5 ±1.1	14.9 ±1.2
	full (accept rule + arbiter)	**41.8 ±1.0**	**27.4 ±1.4**	**29.3 ±0.8**	**22.0 ±0.6**	**15.4 ±1.2**
	executor as probe	42.0 ±1.3	26.6 ±0.8	29.1 ±1.0	21.4 ±1.3	13.8 ±1.1
	τ_r=0.10	41.5 ±1.1	29.3 ±0.9	28.3 ±1.1	21.5 ±0.7	13.5 ±0.8
	executor as probe, τ_r=0.10	42.2 ±1.8	27.9 ±0.7	28.5 ±1.2	22.2 ±0.8	14.1 ±1.0
Gradient (AdamW)	vanilla	60.7 ±0.6	36.0 ±0.7	46.0 ±0.6	35.8 ±0.7	26.9 ±1.2
	drafter, every-step	49.0 ±1.6	41.7 ±1.3	32.1 ±0.6	33.0 ±1.1	24.5 ±1.0
	blind chain	56.7 ±1.5	44.3 ±1.3	33.6 ±0.8	33.0 ±1.1	28.0 ±0.8
	accept rule	55.0 ±1.5	43.1 ±0.9	31.8 ±0.9	33.5 ±0.9	26.3 ±1.1
	full (accept rule + arbiter)	**61.5 ±0.8**	**39.3 ±0.6**	**48.9 ±0.6**	**39.2 ±0.6**	**24.4 ±0.4**
	executor as probe	61.6 ±0.3	39.2 ±0.5	50.1 ±0.6	39.9 ±0.6	25.0 ±0.8
	τ_r=0.10	61.6 ±0.8	45.0 ±0.6	45.8 ±1.2	38.2 ±0.7	25.1 ±1.4
	executor as probe, τ_r=0.10	65.3 ±1.5	46.0 ±0.8	45.4 ±0.8	35.5 ±0.7	26.9 ±0.8

Table 6: **All configurations on Two-Room.** Success rate (%) by goal distance t, with the accept rule's tolerance at its measured value τ=0.127 and the arbiter's at τ_{r}=1.323. Mean over eight evaluation seeds with its standard error. Rows of the full layer are in **bold**; indented rows are its variants and controls (Appendix I).

Executor	Configuration	$t = 25$	$t = 50$	$t = 75$
CEM (LeWM)	vanilla	78.3 ±2.5	60.7 ±1.5	42.2 ±2.6
	drafter, every-step (S=10)	67.4 ±1.8	68.6 ±2.0	56.5 ±1.1
	blind chain	65.4 ±1.8	55.1 ±2.4	47.7 ±2.4
	accept rule	68.2 ±1.8	61.5 ±2.2	53.9 ±2.4
	full (accept rule + arbiter)	**75.2 ±3.1**	**69.5 ±1.4**	**64.1 ±3.0**
	τ_{r}=0.10	68.2 ±1.8	61.5 ±2.2	53.9 ±2.4
GC-IDM	vanilla	100.0 ±0.0	100.0 ±0.0	100.0 ±0.0
	drafter, every-step	99.4 ±0.4	99.8 ±0.2	100.0 ±0.0
	blind chain	97.8 ±0.7	96.7 ±0.8	96.7 ±0.9
	accept rule	98.8 ±0.3	97.7 ±0.9	98.6 ±0.5
	full (accept rule + arbiter)	**100.0 ±0.0**	**100.0 ±0.0**	**100.0 ±0.0**
	executor as probe	100.0 ±0.0	99.8 ±0.2	100.0 ±0.0
	τ_{r}=0.10	99.0 ±0.4	99.0 ±0.4	99.0 ±0.4
	executor as probe, τ_{r}=0.10	98.8 ±0.3	97.7 ±0.9	98.6 ±0.5
iCEM	vanilla	80.5 ±1.9	60.9 ±2.0	40.4 ±2.7
	drafter, every-step	79.7 ±1.6	84.8 ±1.9	81.8 ±1.1
	blind chain	70.5 ±1.7	70.9 ±1.5	73.6 ±1.3
	accept rule	76.2 ±2.0	82.4 ±2.2	79.9 ±1.4
	full (accept rule + arbiter)	**85.7 ±1.7**	**82.2 ±1.5**	**76.4 ±0.8**
	executor as probe	85.9 ±1.5	82.2 ±2.2	80.7 ±1.9
	τ_{r}=0.10	76.2 ±2.0	82.4 ±2.2	79.9 ±1.4
	executor as probe, τ_{r}=0.10	76.2 ±2.0	82.4 ±2.2	79.9 ±1.4
MPPI	vanilla	50.2 ±1.3	45.9 ±2.1	33.0 ±2.0
	drafter, every-step	46.7 ±1.3	35.4 ±2.2	24.2 ±3.0
	blind chain	45.9 ±2.3	28.3 ±2.5	21.5 ±1.6
	accept rule	47.5 ±1.4	32.6 ±1.5	23.8 ±2.5
	full (accept rule + arbiter)	**52.5 ±2.1**	**42.4 ±1.4**	**35.5 ±2.2**
	executor as probe	53.1 ±2.2	44.9 ±1.7	32.8 ±1.1
	τ_{r}=0.10	47.5 ±1.4	32.6 ±1.5	23.8 ±2.5
	executor as probe, τ_{r}=0.10	47.5 ±1.4	32.6 ±1.5	23.8 ±2.5
Gradient (AdamW)	vanilla	84.2 ±1.2	64.3 ±2.8	43.9 ±2.0
	drafter, every-step	72.1 ±1.8	70.3 ±1.6	64.3 ±1.5
	blind chain	71.9 ±1.5	57.8 ±2.1	53.3 ±1.7
	accept rule	72.7 ±2.1	66.4 ±1.6	60.0 ±1.2
	full (accept rule + arbiter)	**82.0 ±1.8**	**72.8 ±1.8**	**60.0 ±2.5**
	executor as probe	83.6 ±1.2	72.5 ±1.6	56.0 ±3.0
	τ_{r}=0.10	72.7 ±2.1	66.4 ±1.6	60.0 ±1.2
	executor as probe, τ_{r}=0.10	72.7 ±2.1	66.4 ±1.6	60.0 ±1.2

B CALIBRATED CONSTANTS

All constants of this section were fixed by offline measurements before the batteries that use them ran; no closed-loop result selected any of them.

B.1 TOLERANCE

Table 7 reports the measurement of Equation (6). The pairs come from different episodes of each environment's offline dataset (held-out episodes on Reacher), Two-Room's accept rule is measured in the DINOv2 half of its paired space and its arbiter in LeWM space, where task-equivalent pairs lie almost as far apart as random ones (median 1.323 against 1.408), and the interval is a percentile bootstrap of the median. Every result uses the median, rounded to three decimals.

An earlier version of our experiments used τ=0.20, the first value we had measured (on PushT), in PushT, Reacher and Cube, and 0.098 on Two-Room. Table 8 shows that mean success moves by at most 1.9 points between the two, except over GC-IDM on Cube (3.0) and iCEM on Two-Room (4.0), whereas the call ratio follows the tolerance.

Table 11 moves the tolerance to either end of its bootstrap interval and to the value of another equivalence a user could adopt (PushT's $5°$ success test; the scored block alone on Cube). On Two-Room it moves the arbiter's tolerance to the readings from pairs within one episode (1.160) and from the agent alone (1.364). Within the interval no cell moves by more than 1.8 points. The alternative readings move PushT and Cube cells by at most 1.5 points and Two-Room cells by at most 5.2, and at either Two-Room reading the arbiter still routes 72% of the t=25 episodes. Table 8 reads the Two-Room arbiter against the accept rule's tolerance, under which it does not act, so its Two-Room columns differ from Table 1, which reads it against 1.323.

Table 7: **Measured tolerance per environment.** The distribution of $\mathrm{rel}(E(o), E(o'))$ over pairs of observations from different episodes that the stated equivalence treats as the same state. The tolerance is the median; the interval is a percentile bootstrap of the median over the pairs.

Environment	Equivalent pairs	pairs	10%	median = τ	90%	95% interval
PushT	agent, block within 20 px; angle within 20°	458	0.036	**0.233**	0.843	[0.207, 0.268]
Reacher	both joints within 0.05 rad	502	0.046	**0.106**	0.153	[0.102, 0.111]
Cube	block and effector within 0.04 m	512	0.418	**0.794**	1.164	[0.759, 0.826]
Two-Room	agent and target within 16 px (DINOv2)	6000	0.058	**0.127**	0.235	[0.125, 0.130]
	the same pairs, LeWM (arbiter)	6000	0.691	**1.323**	1.543	[1.316, 1.329]

Table 8: **Transferred against measured tolerance, full layer.** In each pair of columns the left value is served at the tolerance of the earlier version of the experiments and the right value at the measured tolerance used throughout this paper. On Two-Room both columns read the arbiter against the accept rule's tolerance, under which it does not act; Table 1 reads it against 1.323.

Executor	PushT		Reacher		Cube		Two-Room	
	τ=0.20	τ=0.233	τ=0.20	τ=0.106	τ=0.20	τ=0.794	τ=0.098	τ=0.127
Full layer, mean success rate (%) over the goal distances								
CEM (LeWM)	96.0	95.6	95.1	95.2	46.3	45.8	61.9	61.2
GC-IDM	95.7	96.1	94.4	94.8	60.8	57.8	99.2	98.6
iCEM	92.6	92.3	96.5	96.2	44.6	42.7	83.5	79.5
MPPI	94.5	94.1	91.6	90.9	27.0	27.2	35.9	34.6
Gradient (AdamW)	96.7	96.4	96.6	96.2	43.5	42.7	68.2	66.3
Full layer, call ratio at the longest goal distance								
CEM (LeWM)	0.65	0.58	0.76	0.91	0.99	0.53	0.96	0.88
GC-IDM	0.77	0.73	0.45	0.78	0.94	0.53	0.76	0.71
iCEM	0.72	0.64	0.73	0.89	0.99	0.48	0.93	0.84
MPPI	0.73	0.67	0.95	0.99	0.99	0.56	0.95	0.85
Gradient (AdamW)	0.67	0.60	0.86	0.96	0.99	0.53	0.94	0.85

B.2 DRAFTING BUDGET

The rule of Section 3.5 gives k=3 on PushT and Cube, where the repeat-draw dispersion steps at k=3. On Reacher the dispersion shows no step, so the search executors use k=8 and GC-IDM k=4; the paired Two-Room drafter uses its sampler's default, k=50. Table 9 varies the budget around the served value. One step below the onset of the plateau the layer fails at long range: at k=2 on PushT every draft is rejected and success at t=150 falls from 93%–97% to 53%–67% over the search executors. Above the onset the budget hardly matters: at k=4 and k=8 no PushT or Cube cell moves by more than 3.1 points, and on Reacher no cell is more than 4.7 points from its served value at any budget from k=2 to k=16.

Table 9: **Drafting budget.** Success rate (%) of the full layer, with its call ratio in parentheses, at the shortest and the longest goal distance, for budgets k around the served one, which is marked $*$. Mean over eight evaluation seeds.

PushT	t	k=2	k=3	k=4	k=8
CEM (LeWM)	25	98.0 (1.00)	99.4 (0.88)*	99.6 (0.89)	99.5 (0.87)
	150	60.1 (1.00)	97.1 (0.58)*	97.3 (0.52)	97.9 (0.50)
GC-IDM	25	93.5 (1.00)	99.8 (0.84)*	99.8 (0.82)	99.9 (0.81)
	150	15.9 (1.00)	94.2 (0.73)*	96.2 (0.70)	97.2 (0.69)
iCEM	25	96.7 (1.00)	97.4 (0.88)*	97.3 (0.88)	97.3 (0.88)
	150	53.3 (1.00)	93.1 (0.64)*	95.1 (0.59)	96.2 (0.57)
MPPI	25	94.0 (1.00)	97.1 (0.91)*	96.7 (0.91)	96.8 (0.90)
	150	58.2 (1.00)	94.4 (0.67)*	96.1 (0.60)	97.1 (0.58)
Gradient (AdamW)	25	98.7 (1.00)	99.2 (0.90)*	99.4 (0.89)	99.1 (0.88)
	150	67.0 (1.00)	96.9 (0.60)*	97.6 (0.55)	98.4 (0.53)

Reacher	t	k=2	k=4	k=8	k=16
CEM (LeWM)	25	98.8 (1.00)	98.9 (0.94)	98.2 (0.92)*	98.6 (0.90)
	150	88.7 (1.00)	91.7 (0.94)	90.9 (0.91)*	90.0 (0.90)
GC-IDM	25	97.8 (1.00)	95.3 (0.64)*	94.6 (0.62)	94.5 (0.61)
	150	95.2 (1.00)	95.3 (0.78)*	94.3 (0.76)	94.0 (0.75)
iCEM	25	99.8 (1.00)	98.7 (0.93)	98.8 (0.93)*	98.6 (0.92)
	150	89.8 (1.00)	93.7 (0.93)	91.9 (0.89)*	91.9 (0.89)
MPPI	25	79.6 (1.00)	82.6 (0.99)	81.0 (0.99)*	79.8 (0.99)
	150	90.8 (1.00)	93.3 (0.99)	92.8 (0.99)*	93.0 (0.99)
Gradient (AdamW)	25	96.8 (1.00)	96.9 (0.99)	96.5 (0.94)*	96.5 (0.97)
	150	88.6 (1.00)	94.2 (0.97)	93.3 (0.96)*	93.6 (0.96)

Cube	t	k=2	k=3	k=4	k=8
CEM (LeWM)	25	60.8 (0.99)	66.4 (0.77)*	66.2 (0.76)	66.3 (0.74)
	150	22.4 (1.00)	26.1 (0.53)*	26.7 (0.52)	26.7 (0.53)
GC-IDM	25	70.1 (0.97)	76.7 (0.55)*	77.3 (0.55)	77.2 (0.55)
	150	32.0 (0.97)	45.5 (0.53)*	45.5 (0.55)	48.3 (0.57)
iCEM	25	60.5 (0.98)	62.5 (0.74)*	62.4 (0.73)	62.7 (0.74)
	150	21.7 (1.00)	24.4 (0.48)*	24.3 (0.51)	25.4 (0.52)
MPPI	25	41.6 (0.99)	41.8 (0.76)*	43.3 (0.77)	42.1 (0.77)
	150	11.7 (1.00)	15.4 (0.56)*	15.0 (0.56)	14.2 (0.57)
Gradient (AdamW)	25	59.0 (0.99)	61.5 (0.78)*	62.0 (0.76)	62.9 (0.79)
	150	19.1 (1.00)	24.4 (0.53)*	26.3 (0.53)	25.4 (0.53)

B.3 DRAFTER TRAINING SEED

Every table uses one drafter checkpoint per environment, trained with seed 42. Table 10 repeats the full layer on PushT and Reacher with drafters trained with seeds 1 and 2 under the same recipe; Cube and Two-Room have one checkpoint. No cell moves by more than 2.1 points.

Table 10: **Drafter training seeds.** Success rate (%) of the full layer with the drafter of every other table (seed 42) and with two further drafters, mean over eight evaluation seeds with its standard error.

PushT	drafter seed	t=25	t=50	t=75	t=100	t=150
CEM (LeWM)	42	99.4 ±0.1	93.1 ±0.8	94.9 ±0.5	93.8 ±0.5	97.1 ±0.5
	1	99.3 ±0.2	93.8 ±0.6	94.3 ±0.5	94.9 ±0.5	98.0 ±0.3
	2	99.2 ±0.1	93.1 ±0.3	93.9 ±0.3	94.8 ±0.4	97.9 ±0.3
GC-IDM	42	99.8 ±0.1	96.1 ±0.4	94.9 ±0.3	95.5 ±0.3	94.2 ±0.3
	1	99.9 ±0.1	97.0 ±0.3	96.9 ±0.2	96.1 ±0.4	95.2 ±0.4
	2	99.8 ±0.1	96.0 ±0.3	95.7 ±0.2	95.5 ±0.2	94.6 ±0.3
iCEM	42	97.4 ±0.2	89.7 ±0.8	91.0 ±0.5	90.4 ±0.4	93.1 ±0.4
	1	97.3 ±0.1	90.4 ±0.4	90.9 ±0.4	91.9 ±0.6	94.9 ±0.3
	2	97.3 ±0.3	91.3 ±0.6	91.5 ±0.5	91.9 ±0.2	94.0 ±0.4
MPPI	42	97.1 ±0.4	92.1 ±0.5	92.4 ±0.6	94.5 ±0.7	94.4 ±0.4
	1	96.8 ±0.2	92.5 ±0.4	93.6 ±0.7	95.1 ±0.4	96.5 ±0.3
	2	97.1 ±0.3	91.5 ±0.4	93.1 ±0.5	95.4 ±0.5	95.2 ±0.4
Gradient (AdamW)	42	99.2 ±0.2	94.9 ±0.2	95.6 ±0.4	95.4 ±0.3	96.9 ±0.4
	1	99.1 ±0.2	95.2 ±0.5	94.7 ±0.5	96.6 ±0.5	97.9 ±0.2
	2	99.0 ±0.2	95.4 ±0.5	95.3 ±0.5	95.5 ±0.4	97.4 ±0.2

Reacher	drafter seed	t=25	t=50	t=100	t=150
CEM (LeWM)	42	98.2 ±0.4	97.6 ±0.3	94.0 ±0.5	90.9 ±0.6
	1	98.9 ±0.3	98.6 ±0.1	94.6 ±0.7	91.6 ±0.8
	2	98.3 ±0.7	98.0 ±0.7	93.8 ±0.7	89.7 ±0.7
GC-IDM	42	95.3 ±0.4	93.5 ±0.6	95.0 ±0.4	95.3 ±0.5
	1	95.0 ±0.4	94.1 ±0.7	96.6 ±0.3	94.9 ±0.3
	2	94.2 ±0.3	92.1 ±0.5	95.9 ±0.4	93.9 ±0.2
iCEM	42	98.8 ±0.3	98.2 ±0.2	95.9 ±0.4	91.9 ±0.6
	1	98.5 ±0.4	98.5 ±0.3	96.6 ±0.3	93.2 ±0.4
	2	98.7 ±0.4	98.6 ±0.2	96.6 ±0.3	91.8 ±0.7
MPPI	42	81.0 ±1.1	93.8 ±1.1	96.2 ±0.7	92.8 ±0.8
	1	80.1 ±1.4	95.6 ±0.6	96.8 ±0.6	94.0 ±0.6
	2	79.7 ±1.8	94.5 ±0.4	95.7 ±0.3	91.8 ±0.6
Gradient (AdamW)	42	96.5 ±0.5	98.8 ±0.1	96.4 ±0.3	93.3 ±1.0
	1	96.3 ±0.7	98.9 ±0.3	96.8 ±0.5	93.3 ±0.5
	2	95.1 ±0.6	99.1 ±0.2	96.9 ±0.7	92.3 ±0.6

Table 11: **Sensitivity of the full layer to the tolerance.** Success rate (%) with the call ratio in parentheses, at the shortest and the longest goal distance. The columns marked lower end and upper end bound the 95% bootstrap interval of the measured tolerance; the remaining columns are the other readings of the equivalence described in the text. On Two-Room the columns vary the arbiter's tolerance τ_r, measured in LeWM space, with the accept rule at τ=0.127.

PushT	t	τ=0.173 5° test	τ=0.207 lower end	τ=0.233 measured	τ=0.268 upper end
CEM (LeWM)	25	99.4 (0.92)	99.3 (0.91)	99.4 (0.88)	99.3 (0.89)
	150	96.4 (0.73)	96.7 (0.63)	97.1 (0.58)	96.3 (0.53)
GC-IDM	25	99.6 (0.91)	99.8 (0.86)	99.8 (0.84)	100.0 (0.82)
	150	92.9 (0.81)	93.4 (0.76)	94.2 (0.73)	94.4 (0.70)
iCEM	25	98.4 (0.93)	97.8 (0.90)	97.4 (0.88)	96.5 (0.89)
	150	92.5 (0.81)	92.9 (0.70)	93.1 (0.64)	93.0 (0.57)
MPPI	25	97.1 (0.95)	97.2 (0.92)	97.1 (0.91)	96.9 (0.90)
	150	95.5 (0.79)	94.4 (0.71)	94.4 (0.67)	95.2 (0.61)
Gradient (AdamW)	25	99.3 (0.93)	99.2 (0.91)	99.2 (0.90)	99.0 (0.89)
	150	97.5 (0.75)	97.1 (0.66)	96.9 (0.60)	96.6 (0.55)

Reacher	t	τ=0.102 lower end	τ=0.106 measured	τ=0.111 upper end
CEM (LeWM)	25	98.9 (0.94)	98.2 (0.92)	98.8 (0.92)
	150	91.0 (0.90)	90.9 (0.91)	90.0 (0.90)
GC-IDM	25	93.5 (0.65)	95.3 (0.64)	96.0 (0.62)
	150	95.1 (0.79)	95.3 (0.78)	95.2 (0.76)
iCEM	25	99.2 (0.93)	98.8 (0.93)	99.3 (0.96)
	150	92.3 (0.89)	91.9 (0.89)	91.5 (0.87)
MPPI	25	79.2 (0.99)	81.0 (0.99)	80.8 (1.00)
	150	94.1 (0.99)	92.8 (0.99)	93.8 (0.98)
Gradient (AdamW)	25	95.7 (0.98)	96.5 (0.94)	96.4 (0.94)
	150	92.1 (0.96)	93.3 (0.96)	93.0 (0.95)

Cube	t	τ=0.759 lower end	τ=0.794 measured	τ=0.826 upper end	τ=0.967 block only
CEM (LeWM)	25	66.4 (0.76)	66.4 (0.77)	66.3 (0.75)	66.1 (0.75)
	150	26.3 (0.54)	26.1 (0.53)	26.1 (0.52)	25.1 (0.49)
GC-IDM	25	76.2 (0.56)	76.7 (0.55)	77.0 (0.55)	76.2 (0.56)
	150	45.7 (0.54)	45.5 (0.53)	45.3 (0.52)	45.4 (0.51)
iCEM	25	62.6 (0.73)	62.5 (0.74)	62.4 (0.72)	62.7 (0.70)
	150	24.2 (0.50)	24.4 (0.48)	24.9 (0.47)	24.8 (0.45)
MPPI	25	41.9 (0.77)	41.8 (0.76)	41.8 (0.77)	42.3 (0.75)
	150	15.0 (0.58)	15.4 (0.56)	14.4 (0.55)	15.1 (0.51)
Gradient (AdamW)	25	61.2 (0.76)	61.5 (0.78)	61.5 (0.78)	61.6 (0.77)
	150	24.6 (0.53)	24.4 (0.53)	24.0 (0.52)	22.9 (0.48)

Two-Room	t	τ_r=1.160 same episode	τ_r=1.323 measured	τ_r=1.364 agent only
CEM (LeWM)	25	76.0 (0.94)	75.2 (0.95)	75.2 (0.95)
	75	60.9 (0.88)	64.1 (0.88)	60.4 (0.89)
GC-IDM	25	100.0 (1.00)	100.0 (1.00)	100.0 (1.00)
	75	100.0 (1.00)	100.0 (1.00)	100.0 (1.00)
iCEM	25	86.1 (0.93)	85.7 (0.93)	85.7 (0.93)
	75	76.8 (0.86)	76.4 (0.87)	74.2 (0.88)
MPPI	25	53.3 (0.88)	52.5 (0.88)	52.5 (0.88)
	75	30.9 (0.86)	35.5 (0.86)	35.9 (0.87)
Gradient (AdamW)	25	81.8 (0.94)	82.0 (0.94)	82.0 (0.94)
	75	65.2 (0.85)	60.0 (0.86)	61.1 (0.86)

C THE TWO ENDS OF THE RANGE

Table 12 reports every configuration at the shortest and the longest goal distance of each environment; Figure 1(b) plots its PushT and Reacher columns for the vanilla executor and the full layer, and Section 4.2 discusses the short and the long horizon. Table 2 splits the goal distances into short ($t \leq 50$) and long ($t \geq 75$); its long-range result, that the same seventeen pairs rise beyond two pooled standard errors, holds equally over $t > 25$, the distances beyond the vanilla executor's twenty-five-step lookahead.

Table 12: **The two ends of the range.** Success rate (%) at the shortest and the longest goal distance of each environment, for every executor and configuration. The full row is green where it beats the vanilla executor by more than two pooled standard errors and red where it loses by more; bold marks the best configuration in each block and column. All distances, with standard errors, are in Appendix A.

Executor	Configuration	PushT		Reacher		Cube		Two-Room	
		t=25	t=150	t=25	t=150	t=25	t=150	t=25	t=75
CEM (LeWM)	vanilla	96.4	3.3	95.7	69.6	60.7	20.3	**78.3**	42.2
	+ drafter (every-step)	99.2	95.8	65.2	92.7	59.2	26.0	67.4	56.5
	+ drafter, blind chain	98.5	94.9	50.5	86.2	**69.4**	**32.4**	65.4	47.7
	+ drafter, accept rule	98.8	**97.7**	62.8	**93.1**	66.0	26.8	68.2	53.9
	SURVEYOR (full)	**99.4**	97.1	**98.2**	90.9	66.4	26.1	75.2	**64.1**
GC-IDM	vanilla	**100.0**	62.7	**100.0**	**97.7**	71.5	31.6	**100.0**	**100.0**
	+ drafter (every-step)	99.3	94.9	80.6	95.3	74.9	**51.2**	99.4	**100.0**
	+ drafter, blind chain	99.2	**95.7**	54.1	93.2	80.7	51.1	97.8	96.7
	+ drafter, accept rule	99.4	94.0	60.7	94.6	**81.4**	49.5	98.8	98.6
	SURVEYOR (full)	99.8	94.2	95.3	95.3	76.7	45.5	**100.0**	**100.0**
iCEM	vanilla	97.6	1.8	97.4	74.1	61.3	20.4	80.5	40.4
	+ drafter (every-step)	99.0	92.1	61.5	93.1	65.8	28.4	79.7	**81.8**
	+ drafter, blind chain	**99.0**	92.6	42.8	86.9	**68.0**	**31.4**	70.5	73.6
	+ drafter, accept rule	**99.0**	**93.9**	53.0	**93.1**	67.2	28.6	76.2	79.9
	SURVEYOR (full)	97.4	93.1	**98.8**	91.9	62.5	24.4	**85.7**	76.4
MPPI	vanilla	91.5	4.2	81.0	81.5	**41.8**	11.3	50.2	33.0
	+ drafter (every-step)	97.6	94.8	54.2	91.7	30.4	11.7	46.7	24.2
	+ drafter, blind chain	96.4	87.1	47.8	89.2	32.4	14.1	45.9	21.5
	+ drafter, accept rule	**97.7**	**94.9**	52.4	92.5	32.3	14.9	47.5	23.8
	SURVEYOR (full)	97.1	94.4	**81.0**	**92.8**	**41.8**	**15.4**	**52.5**	**35.5**
Gradient (AdamW)	vanilla	98.1	8.8	**97.2**	89.3	60.7	26.9	**84.2**	43.9
	+ drafter (every-step)	98.8	**96.9**	66.8	93.3	49.0	24.5	72.1	**64.3**
	+ drafter, blind chain	98.5	95.7	52.2	89.7	56.7	**28.0**	71.9	53.3
	+ drafter, accept rule	98.7	96.8	63.6	**93.5**	55.0	26.3	72.7	60.0
	SURVEYOR (full)	**99.2**	**96.9**	96.5	93.3	**61.5**	24.4	82.0	60.0

D WHAT LIMITS THE LIFT

D.1 SHORT-RANGE COMPETENCE AND THE LIFT

The layer turns one long problem into several short ones, so its lift tracks what the executor can do at short range. At t=25 the vanilla search executors solve 91%–98% of PushT and 81%–97% of Reacher episodes but only 42%–61% of Cube episodes, and under the full layer at the longest distance they reach 93%–97%, 91%–93% and 15%–26% (Table 12). GC-IDM, the strongest short-range executor on Cube, receives the largest lift there, and on Two-Room the smallest lift goes to MPPI, the weakest executor there at short range. This evidence is correlational.

D.2 CUTTING THE EXECUTOR'S SEARCH BUDGET

Table 13 cuts the search budget of CEM and iCEM on PushT and of CEM on Reacher from 300 samples and 30 iterations to two samples and one iteration, with the drafter, the episodes and the seeds unchanged. Success under every-step drafting at t=150 follows the ten-step planner's success at t=25 level by level. On PushT the dependence is steep: over CEM a loss of 20 points at short range costs 77 at long range, and at the level that matches Cube's search executors (60.9% at t=25) every-step drafting solves under 1% of the episodes at t=150. On Reacher it is shallower, 26.5% at t=150 where the planner solves 13.4% at t=25. The intervention weakens the executors of two other environments, not Cube's own, so its bearing on Cube is an inference.

Table 13: **The executor's search budget cut with everything else held fixed.** Success rate (%) of the vanilla ten-step planner at t=25 and of every-step drafting over the same planner at t=150 (the two columns of each group), with the drafter, the populations and the seeds unchanged. Mean over eight evaluation seeds with its standard error.

Search budget	PushT, CEM		PushT, iCEM		Reacher, CEM	
samples × iterations	t=25	t=150	t=25	t=150	t=25	t=150
300 × 30 (served)	95.8 ±0.5	95.8 ±0.6	91.2 ±0.1	92.1 ±0.4	95.7 ±0.3	92.7 ±0.7
100 × 10	94.1 ±0.2	96.6 ±0.4	89.8 ±0.2	90.7 ±0.5	94.9 ±0.3	93.1 ±0.3
30 × 5	88.0 ±0.5	94.1 ±0.2	86.6 ±0.2	88.6 ±0.7	94.9 ±0.3	93.4 ±0.5
10 × 3	81.4 ±0.8	74.3 ±0.5	80.6 ±0.6	62.3 ±0.8	87.9 ±0.5	90.9 ±0.7
10 × 1	75.4 ±0.4	18.8 ±0.8	72.8 ±0.2	12.8 ±0.6	73.6 ±1.1	83.6 ±0.7
5 × 1	66.9 ±0.4	3.8 ±0.2	65.7 ±0.5	3.1 ±0.2	47.1 ±1.3	59.6 ±1.1
2 × 1	60.9 ±0.4	0.8 ±0.3	59.3 ±0.3	0.2 ±0.1	13.4 ±0.9	26.5 ±1.3

D.3 DEMONSTRATION FRAMES AS SUBGOALS

Table 14 replaces the drafter by frames of the demonstration that produced the goal, spaced S=10 steps apart, served either on a fixed schedule or under the accept rule. Neither is an upper bound on the layer: scheduled frames run ahead of a lagging executor and trail the drafter by 12 to 37 points at PushT t=150, and frames under the accept rule stall whenever the executor reaches the region by another path, since nothing re-drafts them from the achieved state. Two readings survive. On Reacher at t=100 and t=150 scheduled frames reach 98%–100% against the drafter's 92%–95%, so the drafter leaves success on the table there. On Cube at t=150 they stay below 46%, so better subgoals alone would not bring Cube near PushT or Reacher.

E BEYOND t=150

Table 15 extends the goal distance as far as the data allow: t=200 on PushT, whose longest demonstration has 246 steps, and t=195 on Reacher and Cube, whose episodes have 201. The populations are every held-out episode long enough on PushT (197 at t=175, 50 at t=200) and 128 held-out pairs on Reacher and Cube. GC-IDM is absent because no checkpoint exists for step budgets of 350 or more. The lift persists on PushT and Reacher, whereas on Cube every configuration collapses; Figure 4 plots these distances as hollow markers.

Table 14: **Demonstration frames as subgoals.** Success rate (%) of each search executor when vanilla, under every-step drafting, and when the drafter is replaced by frames of the demonstration that produced the goal, served either on a fixed schedule or under the accept rule. Mean over eight evaluation seeds. Neither demonstration row is an upper bound (see the text).

Executor	Subgoals	PushT			Reacher			Cube		
		t=50	t=100	t=150	t=50	t=100	t=150	t=50	t=100	t=150
CEM	none (vanilla)	50.6	13.7	3.3	93.3	81.3	69.6	38.4	35.1	20.3
	drafter, every-step	95.3	94.0	95.8	88.3	95.2	92.7	43.4	36.6	26.0
	demonstration, on schedule	97.3	91.1	83.8	99.2	99.8	100.0	64.2	40.3	44.2
	demonstration, accept rule	69.6	45.5	31.5	42.5	26.0	25.2	41.3	29.9	15.7
iCEM	none (vanilla)	47.0	11.9	1.8	96.8	82.4	74.1	40.4	34.3	20.4
	drafter, every-step	92.3	88.0	92.1	83.8	93.3	93.1	46.5	35.8	28.4
	demonstration, on schedule	96.3	90.2	73.0	98.2	98.6	100.0	54.6	30.7	39.5
	demonstration, accept rule	72.6	40.8	26.1	66.1	58.3	57.7	34.0	24.7	14.0
MPPI	none (vanilla)	51.3	12.5	4.2	90.4	83.0	81.5	25.3	20.7	11.3
	drafter, every-step	93.5	94.3	94.8	76.0	93.0	91.7	25.7	18.5	11.7
	demonstration, on schedule	89.9	73.8	57.7	90.1	98.2	99.6	24.1	18.6	13.3
	demonstration, accept rule	60.2	31.8	18.4	16.0	10.6	11.6	22.4	18.4	10.2
Gradient	none (vanilla)	72.4	21.9	8.8	94.1	92.8	89.3	36.0	35.8	26.9
	drafter, every-step	95.9	94.8	96.9	88.1	95.3	93.3	41.7	33.0	24.5
	demonstration, on schedule	96.6	89.0	81.2	98.5	99.9	100.0	56.2	40.0	45.8
	demonstration, accept rule	69.0	38.2	25.1	20.0	12.1	12.4	35.9	32.4	20.8

Table 15: **Beyond** t=150**.** Success rate (%) at each environment's measured tolerance, with the t=150 cell of the same configuration for reference. The vanilla executor is the better of its two horizons per cell.

Executor	Configuration	PushT			Reacher			Cube		
		t=150	t=175	t=200	t=150	t=175	t=195	t=150	t=175	t=195
CEM (LeWM)	vanilla	3.3 ±0.2	3.1 ±0.4	2.0 ±1.0	69.6 ±0.5	68.6 ±0.4	66.3 ±0.5	20.3 ±0.7	2.3 ±0.5	0.2 ±0.1
	every-step	95.8 ±0.6	96.3 ±0.5	96.5 ±1.1	92.7 ±0.7	93.1 ±0.7	89.1 ±0.8	26.0 ±1.6	8.1 ±0.5	2.7 ±0.5
	accept rule	97.7 ±0.4	96.4 ±0.4	98.5 ±0.5	93.1 ±0.5	91.9 ±0.5	88.8 ±0.6	26.8 ±1.0	7.5 ±0.8	2.5 ±0.6
	full layer	97.1 ±0.5	97.0 ±0.4	99.0 ±0.5	90.9 ±0.6	91.0 ±0.6	86.6 ±0.6	26.1 ±0.8	11.4 ±1.3	1.4 ±0.4
iCEM	vanilla	1.8 ±0.1	0.8 ±0.2	0.0 ±0.0	74.1 ±0.4	71.4 ±0.5	67.3 ±0.3	20.4 ±0.5	2.7 ±0.3	0.0 ±0.0
	every-step	92.1 ±0.4	94.3 ±0.8	96.2 ±1.3	93.1 ±0.6	93.5 ±0.6	89.9 ±0.6	28.4 ±0.7	11.6 ±0.9	0.3 ±0.1
	accept rule	93.9 ±0.4	96.2 ±0.4	97.2 ±0.5	93.1 ±0.5	93.5 ±0.5	88.4 ±0.4	28.6 ±0.5	9.9 ±0.8	0.5 ±0.2
	full layer	93.1 ±0.4	96.6 ±0.4	98.2 ±0.7	91.9 ±0.6	93.5 ±0.6	88.0 ±1.1	24.4 ±0.8	9.4 ±0.4	0.2 ±0.1
MPPI	vanilla	4.2 ±0.4	3.2 ±0.4	3.5 ±1.1	81.5 ±0.8	83.0 ±0.7	79.9 ±1.0	11.3 ±0.6	1.6 ±0.2	0.2 ±0.1
	every-step	94.8 ±0.3	96.0 ±0.5	91.2 ±1.2	91.7 ±0.4	93.1 ±0.6	89.2 ±0.5	11.7 ±1.0	4.9 ±0.8	2.7 ±0.4
	accept rule	94.9 ±0.5	96.8 ±0.5	93.8 ±0.9	92.5 ±0.6	92.9 ±0.5	88.8 ±0.6	14.9 ±1.2	6.5 ±0.3	5.0 ±0.5
	full layer	94.4 ±0.4	96.4 ±0.4	93.5 ±0.6	92.8 ±0.8	93.4 ±0.6	89.7 ±0.5	15.4 ±1.2	5.5 ±0.6	2.5 ±0.6
Gradient (AdamW)	vanilla	8.8 ±0.5	9.4 ±0.5	7.2 ±1.5	89.3 ±0.7	88.0 ±0.7	85.0 ±0.6	26.9 ±1.2	3.9 ±0.5	0.2 ±0.1
	every-step	96.9 ±0.5	96.4 ±0.3	93.0 ±1.6	93.3 ±0.8	93.4 ±0.3	88.3 ±0.9	24.5 ±1.0	5.2 ±0.4	3.2 ±0.6
	accept rule	96.8 ±0.3	97.5 ±0.2	97.5 ±0.7	93.5 ±0.6	93.8 ±0.4	88.2 ±0.6	26.3 ±1.1	5.8 ±0.6	3.1 ±0.5
	full layer	96.9 ±0.4	96.8 ±0.5	98.0 ±0.8	93.3 ±1.0	93.4 ±1.0	87.8 ±0.8	24.4 ±0.4	7.4 ±0.6	1.8 ±0.3

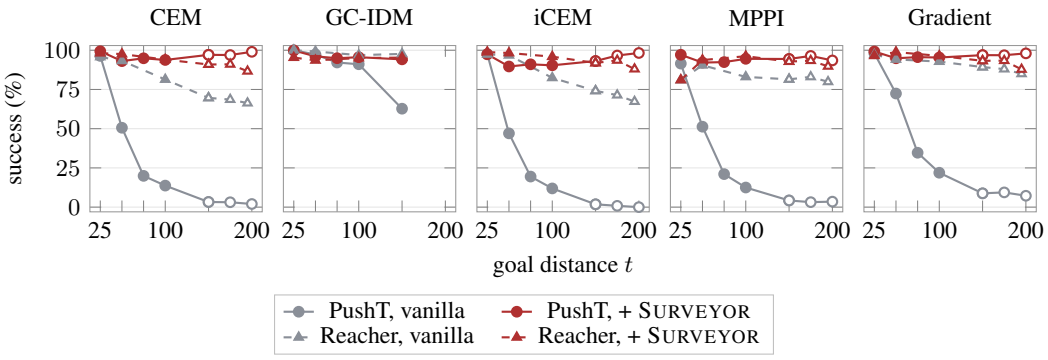

Figure 4: **Success against goal distance.** Each point is the mean over eight evaluation seeds with its standard error. On PushT the vanilla executor falls with distance whereas the full layer stays above 89% throughout; beyond two pooled standard errors it falls below the vanilla executor only over GC-IDM, by at most 1.2 points on PushT at t=25 and t=50 and by 1.9 to 5.7 points on Reacher at every distance. Hollow markers, beyond t=150, use held-out populations, smaller on PushT (Appendix E).

F COMPONENT ABLATION

Table 16 reports the intermediate configurations that Table 1 omits: every-step drafting, the blind chain and the accept rule, each over every executor, beside the full layer; Sections 4.2 to 4.4 discuss them in turn.

Table 16: **Component ablation.** For each executor, the mean / worst success rate (%) over the nine PushT and Reacher settings under every-step drafting, the blind chain, the accept rule and the full layer, and the call ratio of each at PushT t=150; every-step drafting is 1.00 by definition. Per-setting values are in Appendix A.

Executor	success over PushT + Reacher, mean / worst				call ratio at PushT t=150			
	every-step	blind chain	accept rule	full	every-step	blind chain	accept rule	full
CEM (LeWM)	91.2 / 65.2	85.0 / 50.5	90.8 / 62.8	**95.4 / 90.9**	1.00	0.35	0.58	**0.58**
GC-IDM	94.2 / 80.6	87.6 / 54.1	90.3 / 60.7	**95.5 / 93.5**	1.00	0.33	0.73	**0.73**
iCEM	88.1 / 61.5	83.2 / 42.8	87.7 / 53.0	**94.0 / 89.7**	1.00	0.35	0.63	**0.64**
MPPI	87.6 / 54.2	81.7 / 47.8	88.0 / 52.4	**92.7 / 81.0**	1.00	0.34	0.67	**0.67**
Gradient (AdamW)	91.6 / 66.8	86.7 / 52.2	91.4 / 63.6	**96.3 / 93.3**	1.00	0.35	0.60	**0.60**

G What the arbiter does

Table 17 reports how often the full layer's arbiter routes an episode to the executor at the first boundary and how often it retires the drafter later. At t=25 it mostly routes whole episodes: 91% on Reacher, 85% on Cube, 72% on Two-Room and 63% on PushT. At the longest distances routing falls to 0%–39% and the work shifts to retirement, for 12%–14% of PushT episodes and about three fifths of Reacher and Two-Room and two thirds of Cube episodes over the search executors. The Cube losses over GC-IDM at t=50 and t=75 (Section 4.4) come from the 57% and 40% of episodes routed to GC-IDM alone there; the supplementary `arbiter_activity.csv` lists every distance.

Table 17: **What the arbiter does.** Percentage of episodes that the full layer's arbiter routed to the executor alone at the first replanning boundary, and percentage in which it retired the drafter later, at the shortest and the longest goal distance; means over eight evaluation seeds. The probe is computed by the shared CEM planner, so the routing rate is the same for every search executor and within three points of it for GC-IDM, whose retirement is logged as arrival at the goal.

Executor	Arbiter decision	PushT		Reacher		Cube		Two-Room	
		t=25	t=150	t=25	t=150	t=25	t=150	t=25	t=75
CEM (LeWM)	routed at start	63	0	91	39	85	17	72	34
	retired later	16	13	8	61	12	68	19	56
GC-IDM (released code)	routed at start	63	0	91	39	84	15	72	34
	retired later	0	0	1	15	9	38	28	66
iCEM	routed at start	63	0	91	39	85	17	72	34
	retired later	15	14	8	60	13	65	21	62
MPPI	routed at start	63	0	91	39	85	17	72	34
	retired later	16	13	8	61	12	68	19	51
Gradient (AdamW)	routed at start	63	0	91	39	85	17	72	34
	retired later	15	12	8	61	13	69	23	58

H Cost

H.1 Wall-clock and the drafter's share of it

Wall-clock is the scheduler's elapsed time per episode on one NVIDIA A100, model loading included, averaged over tasks that cover all eight seeds; successful episodes end early, so a row that succeeds more often can also be faster. Rows computed on a second cluster were replayed on the first for timing, and a vanilla row is timed at the horizon whose success it prints. Under every-step drafting at t=150, frame encoding and drafting take at most 1.4% of the time spent on drafting and search, so with this drafter the accept rule's saving in drafter calls is not visible in wall-clock; it would matter in proportion to a heavier drafter. Reuse can cut drafter evaluations by at most a factor of N: with independent checks that reject with probability q, the long-run drafting cost per replanning decision is $kq/\big(1-(1-q)^N\big)$ for $q>0$ and k/N at $q=0$. Tables 18 and 19 list every configuration at the longest goal distance.

H.2 Cost per solved episode

Table 20 divides the seconds per episode by the success rate at the longest goal distance. The layer lowers the cost per solved episode in thirteen of the twenty pairs: by 12 to 63 times over the search executors on PushT, by about a third over CEM, iCEM and MPPI on Reacher, and in five of the ten Cube and Two-Room pairs. Against every-step drafting it takes longer per episode in eighteen of the twenty pairs, because the probe costs more than the accept rule's saving in drafter calls returns; what the probe buys is the short-range success of Section 4.4.

Table 21 repeats the comparison at t=25. The layer is then the cheaper of the two in four of the twenty pairs: over the search executors it costs 0.7 to 1.4 times the vanilla executor per solved episode on

Table 18: **Cost of every configuration on PushT and Reacher.** At the longest goal distance (t=150): success rate (SR, %), call ratio and wall-clock seconds per episode (s/ep) on one NVIDIA A100.

Executor	Configuration	PushT $t = 150$			Reacher $t = 150$		
		SR	call ratio	s/ep	SR	call ratio	s/ep
CEM (LeWM)	vanilla	3.3	0	14.9	69.6	0	4.1
	drafter, every-step (S=10)	95.8	1.00	7.8	92.7	1.00	3.0
	drafter, every-step (S=25)	92.8	1.00	9.7	89.7	1.00	3.9
	blind chain (2 waypoints)	97.2	0.51	7.9	91.4	0.52	3.5
	blind chain	94.9	0.35	8.4	86.2	0.36	4.1
	accept rule	97.7	0.58	8.1	93.1	0.81	3.5
	full (accept rule + arbiter)	97.1	0.58	12.1	90.9	0.91	3.2
GC-IDM	vanilla	62.7	0	2.1	97.7	0	1.6
	drafter, every-step	94.9	1.00	1.3	95.3	1.00	1.1
	blind chain	95.7	0.33	1.3	93.2	0.33	1.2
	accept rule	94.0	0.73	1.5	94.6	0.79	1.4
	full (accept rule + arbiter)	94.2	0.73	2.1	95.3	0.78	1.8
iCEM	vanilla	1.8	0	16.0	74.1	0	3.7
	drafter, every-step	92.1	1.00	8.9	93.1	1.00	3.0
	blind chain (2 waypoints)	95.0	0.51	8.5	91.7	0.52	3.5
	blind chain	92.6	0.35	8.7	86.9	0.36	4.1
	accept rule	93.9	0.63	8.8	93.1	0.76	3.6
	full (accept rule + arbiter)	93.1	0.64	13.1	91.9	0.89	2.9
MPPI	vanilla	4.2	0	15.8	81.5	0	4.6
	drafter, every-step	94.8	1.00	8.0	91.7	1.00	3.4
	blind chain (2 waypoints)	93.6	0.51	8.4	91.0	0.52	3.7
	blind chain	87.1	0.34	9.3	89.2	0.35	4.2
	accept rule	94.9	0.67	8.5	92.5	0.97	3.6
	full (accept rule + arbiter)	94.4	0.67	12.4	92.8	0.99	3.6
Gradient (AdamW)	vanilla	8.8	0	50.4	89.3	0	9.5
	drafter, every-step	96.9	1.00	41.8	93.3	1.00	14.1
	blind chain (2 waypoints)	98.0	0.51	43.4	92.9	0.52	16.9
	blind chain	95.7	0.35	45.3	89.7	0.36	19.9
	accept rule	96.8	0.60	43.0	93.5	0.91	14.0
	full (accept rule + arbiter)	96.9	0.60	46.3	93.3	0.96	10.3

PushT, Reacher and Cube, and 1.1 to 3.3 times on Two-Room; over GC-IDM, the cheapest executor, 1.5 to 4.2 times. At Reacher t=25 every-step drafting costs 1.3 to 2.1 times as much per solved episode as the full layer over every search executor.

I ARBITER VARIANTS

I.1 PROBE PLANNER AND RETIREMENT TOLERANCE

The arbiter has two settings: which planner computes the probe, and the retirement tolerance. Table 1 uses the CEM probe and $\tau_{\mathrm{r}} = \tau$; the indented rows of Appendix A compute the probe with the block's own executor ("executor as probe"), retire at τ_{r}=0.10, or both. Over CEM the stricter tolerance repairs PushT t=50 (93.1% to 96.1%) and moves no Reacher setting by more than 1.4 points. Over iCEM every variant raises the mean over PushT and Reacher, from 94.0% to 95.3%–95.6%, and the combination is no better than either setting alone. Over GC-IDM its own probe fails to route the short Reacher episodes (64.3% against 95.3% at t=25), whereas on Cube the combined setting recovers the losses of Section 4.4. No variant is best on every executor and the two settings do not add, so the main text deploys one uniform setting: choosing the best variant per executor from these closed-loop results would be selection on the test cells.

Table 19: **Cost of every configuration on Cube and Two-Room.** At the longest goal distance (t=150 and t=75): success rate (SR, %), call ratio and wall-clock seconds per episode (s/ep) on one NVIDIA A100.

Executor	Configuration	Cube $t = 150$			Two-Room $t = 75$		
		SR	call ratio	s/ep	SR	call ratio	s/ep
CEM (LeWM)	vanilla	20.3	0	9.5	42.2	0	7.0
	drafter, every-step (S=10)	26.0	1.00	7.7	56.5	1.00	7.2
	blind chain	32.4	0.34	7.2	47.7	0.34	7.7
	accept rule	26.8	0.49	11.1	53.9	0.88	7.7
	full (accept rule + arbiter)	26.1	0.53	12.8	64.1	0.88	9.1
GC-IDM	vanilla	31.6	0	2.6	100.0	0	0.7
	drafter, every-step	51.2	1.00	2.3	100.0	1.00	0.6
	blind chain	51.1	0.33	2.3	96.7	0.33	1.0
	accept rule	49.5	0.48	2.8	98.6	0.57	1.7
	full (accept rule + arbiter)	45.5	0.53	2.9	100.0	1.00	2.5
iCEM	vanilla	20.4	0	8.5	40.4	0	7.0
	drafter, every-step	28.4	1.00	7.5	81.8	1.00	5.8
	blind chain	31.4	0.34	7.3	73.6	0.36	6.8
	accept rule	28.6	0.48	10.6	79.9	0.84	6.5
	full (accept rule + arbiter)	24.4	0.48	13.0	76.4	0.87	8.0
MPPI	vanilla	11.3	0	12.9	33.0	0	8.1
	drafter, every-step	11.7	1.00	10.1	24.2	1.00	8.5
	blind chain	14.1	0.34	10.0	21.5	0.34	8.5
	accept rule	14.9	0.51	12.3	23.8	0.85	8.9
	full (accept rule + arbiter)	15.4	0.56	14.1	35.5	0.86	11.4
Gradient (AdamW)	vanilla	26.9	0	23.3	43.9	0	25.6
	drafter, every-step	24.5	1.00	35.0	64.3	1.00	25.2
	blind chain	28.0	0.34	32.6	53.3	0.35	27.1
	accept rule	26.3	0.48	35.1	60.0	0.85	25.6
	full (accept rule + arbiter)	24.4	0.53	37.0	60.0	0.86	25.6

Table 20: **Cost per solved episode at the longest goal distance.** Each executor X, vanilla and under the full layer (X + SURVEYOR), in each environment: success rate (SR, %), wall-clock seconds per episode (s/ep) and seconds per solved episode (s/solved, the seconds per episode divided by the success rate) on one NVIDIA A100. Bold marks the lower cost per solved episode of each pair.

Executor	PushT t=150			Reacher t=150			Cube t=150			Two-Room t=75		
	SR	s/ep	s/solved	SR	s/ep	s/solved	SR	s/ep	s/solved	SR	s/ep	s/solved
CEM	3.3	14.9	457	69.6	4.1	5.9	20.3	9.5	**46.7**	42.2	7.0	16.6
CEM + SURVEYOR	97.1	12.1	**12.5**	90.9	3.2	**3.5**	26.1	12.8	48.9	64.1	9.1	**14.2**
GC-IDM	62.7	2.1	3.3	97.7	1.6	**1.6**	31.6	2.6	8.1	100.0	0.7	**0.7**
GC-IDM + SURVEYOR	94.2	2.1	**2.2**	95.3	1.8	1.9	45.5	2.9	**6.5**	100.0	2.5	2.5
iCEM	1.8	16.0	888	74.1	3.7	4.9	20.4	8.5	**41.9**	40.4	7.0	17.3
iCEM + SURVEYOR	93.1	13.1	**14.1**	91.9	2.9	**3.2**	24.4	13.0	53.2	76.4	8.0	**10.5**
MPPI	4.2	15.8	371	81.5	4.6	5.7	11.3	12.9	114	33.0	8.1	**24.6**
MPPI + SURVEYOR	94.4	12.4	**13.1**	92.8	3.6	**3.9**	15.4	14.1	**91.1**	35.5	11.4	31.9
Gradient	8.8	50.4	573	89.3	9.5	**10.6**	26.9	23.3	**86.5**	43.9	25.6	58.3
Gradient + SURVEYOR	96.9	46.3	**47.8**	93.3	10.3	11.0	24.4	37.0	152	60.0	25.6	**42.6**

I.2 TWO CONTROLS

Over CEM and iCEM, "every-step under the arbiter" keeps the arbiter and replaces the accept rule by every-step drafting. It equals the full layer on Reacher within 1.7 points and trails it on PushT beyond t=25, by 0.8 to 1.9 points over CEM and 2.4 to 4.3 over iCEM, so under the arbiter the accept rule is

Table 21: **Cost per solved episode at the shortest goal distance.** Each executor X, vanilla and under the full layer (X + SURVEYOR), in each environment: success rate (SR, %), wall-clock seconds per episode (s/ep) and seconds per solved episode (s/solved) on one NVIDIA A100. Bold marks the lower cost per solved episode of each pair.

Executor	PushT t=25			Reacher t=25			Cube t=25			Two-Room t=25		
	SR	s/ep	s/solved	SR	s/ep	s/solved	SR	s/ep	s/solved	SR	s/ep	s/solved
CEM	96.4	1.8	1.8	95.7	1.0	**1.0**	60.7	3.3	5.4	78.3	1.2	**1.5**
CEM + SURVEYOR	99.4	1.3	**1.4**	98.2	1.4	1.4	66.4	3.2	**4.8**	75.2	3.8	5.0
GC-IDM	100.0	0.4	**0.4**	100.0	0.7	**0.7**	71.5	1.6	**2.2**	100.0	0.4	**0.4**
GC-IDM + SURVEYOR	99.8	0.7	0.7	95.3	1.1	1.2	76.7	2.5	3.3	100.0	1.8	1.8
iCEM	97.6	1.6	1.7	97.4	1.2	**1.3**	61.3	2.7	**4.4**	80.5	2.1	**2.6**
iCEM + SURVEYOR	97.4	1.4	**1.5**	98.8	1.4	1.4	62.5	3.3	5.3	85.7	4.2	4.8
MPPI	91.5	1.1	**1.2**	81.0	1.6	**2.0**	41.8	3.2	**7.7**	50.2	2.5	**5.0**
MPPI + SURVEYOR	97.1	1.4	1.5	81.0	1.7	2.1	41.8	3.4	8.1	52.5	3.9	7.5
Gradient	98.1	5.5	5.6	97.2	4.1	**4.2**	60.7	5.8	**9.6**	84.2	7.7	**9.1**
Gradient + SURVEYOR	99.2	4.7	**4.7**	96.5	4.3	4.5	61.5	6.3	10.2	82.0	8.5	10.3

not only a saving in drafter calls. "Plan-free router" replaces the probe by the latent distance between the current state and the goal, read against the same tolerance. It almost never routes: its rows equal the accept-rule rows within two pooled standard errors, reading 63.0% and 52.5% at Reacher t=25 where the probe reads 98.2% and 98.8%. The plan is what makes routing possible.

I.3 PRE-REGISTERED PREDICTIONS

Every battery behind these rows was pre-registered with predictions and falsifiers, first run at the tolerance of the earlier version of our experiments (Appendix B) and re-run at the measured tolerances; five of the re-runs' six predictions held, the exception concerning the GC-IDM reimplementation (Appendix J). Of the fifteen predictions on the arbiter variants, five held, three were falsified and seven failed without their falsifiers firing, among them that the two settings add. The falsified three are that a matched probe costs MPPI at least 5 points at Reacher t=25 (it gains 1.7), that GC-IDM's own probe restores Reacher, and that it costs nothing at Cube t=150 (it costs 5.2 points). The pre-registered rule provided that each executor would then deploy its winning single setting; we depart from it and deploy one uniform setting, for the reason given above. Of the two predictions on the controls, the one on every-step drafting under the arbiter held and the one on the plan-free router failed. The Two-Room re-runs at the arbiter's LeWM-space tolerance registered three predictions. Two held: the executor-as-probe rows stay within three points of the CEM probe in ten of twelve cells, and both alternative tolerances still route most t=25 episodes. The third, that no cell moves by more than five points between tolerances, failed by 0.2 points without its falsifier (ten points) firing.

J OUR GC-IDM REIMPLEMENTATION

Table 1 uses the authors' released GC-IDM code; Table 22 repeats every configuration over our own reimplementation as a replication check. The two agree within three points on every PushT, Reacher and Two-Room setting. On Cube the reimplementation is the stronger executor, vanilla by 2.2 to 20.5 points and under the full layer by up to 16, a gap that does not depend on the tolerance.

K LEFLOW

LeFlow (Huang et al., 2026a) amortizes planning into a rectified-flow latent path; we serve the authors' public code, trained on the frozen LeWM at their defaults, with each drafted waypoint handed to it as its target. Table 23 shows that the layer lowers success against the better of LeFlow's two horizons at every distance but Cube t=150, a tie. At PushT t=150 only the blind chain helps, 68.9% against 24.4%, while every-step drafting and the accept rule stay below 12%. At the ten-step

Table 22: **All configurations over our GC-IDM reimplementation.** Success rate (%) by goal distance t, mean over eight evaluation seeds with its standard error.

PushT	Configuration	$t = 25$	$t = 50$	$t = 75$	$t = 100$	$t = 150$
GC-IDM (ours)	vanilla	99.0 ±0.1	96.6 ±0.1	94.6 ±0.1	93.8 ±0.1	61.1 ±0.3
	drafter, every-step	99.3 ±0.1	97.5 ±0.2	95.9 ±0.3	95.7 ±0.3	93.0 ±0.6
	blind chain	99.3 ±0.1	97.0 ±0.2	96.1 ±0.3	96.3 ±0.3	92.9 ±0.6
	accept rule	99.4 ±0.1	97.8 ±0.2	96.3 ±0.4	96.9 ±0.2	93.8 ±0.4
	full (accept rule + arbiter)	**99.6 ±0.0**	**97.7 ±0.3**	**96.8 ±0.4**	**96.0 ±0.3**	**93.1 ±0.6**

Reacher	Configuration	$t = 25$	$t = 50$	$t = 100$	$t = 150$
GC-IDM (ours)	vanilla	100.0 ±0.0	99.2 ±0.0	96.1 ±0.0	96.1 ±0.0
	drafter, every-step	79.8 ±1.8	95.8 ±0.5	96.4 ±0.2	95.6 ±0.2
	blind chain	53.7 ±1.1	70.8 ±1.7	89.3 ±0.9	93.8 ±0.4
	accept rule	60.2 ±1.6	85.2 ±0.9	94.1 ±0.6	94.5 ±0.4
	full (accept rule + arbiter)	**95.3 ±0.4**	**93.5 ±0.8**	**95.1 ±0.6**	**95.6 ±0.4**

Cube	Configuration	$t = 25$	$t = 50$	$t = 75$	$t = 100$	$t = 150$
GC-IDM (ours)	vanilla	73.7 ±0.2	61.9 ±0.3	58.3 ±0.3	70.4 ±0.3	38.5 ±0.7
	drafter, every-step	79.1 ±0.9	71.1 ±1.1	63.3 ±0.9	73.0 ±0.4	50.0 ±0.8
	blind chain	85.8 ±0.3	86.0 ±0.6	66.4 ±1.1	74.8 ±0.6	53.7 ±0.6
	accept rule	83.4 ±0.4	76.8 ±1.4	63.9 ±0.5	73.2 ±0.5	50.0 ±0.3
	full (accept rule + arbiter)	**77.5 ±0.5**	**68.8 ±0.8**	**63.9 ±0.6**	**75.0 ±0.3**	**48.7 ±1.0**

Two-Room	Configuration	$t = 25$	$t = 50$	$t = 75$
GC-IDM (ours)	vanilla	100.0 ±0.0	99.8 ±0.2	100.0 ±0.0
	drafter, every-step	99.6 ±0.4	99.8 ±0.2	100.0 ±0.0
	blind chain	97.8 ±0.7	97.5 ±0.6	96.9 ±0.7
	accept rule	97.7 ±0.5	97.3 ±1.0	99.4 ±0.6
	full (accept rule + arbiter)	**99.2 ±0.3**	**98.8 ±0.4**	**100.0 ±0.0**

horizon between subgoals, vanilla LeFlow already loses 11.5 to 56.8 points at t=25 against its twenty-five-step horizon, so this block is reported here rather than in Table 1.

L HI-LEWM AT ITS OWN PROTOCOL

We ran the released code and checkpoints of Hi-LeWM (Caselli et al., 2026) at their own protocol (50 episodes per cell, step budget $2d$, seeds 42 to 44) on PushT and Cube. On PushT their flat LeWM baseline reaches 89.3%, 44.7% and 17.3% at $d = 25$, 50 and 75 (their paper reports 94.0%, 52.7% and 18.0%), and no hierarchical configuration exceeds it at $d = 25$ or 50; at $d = 75$ the best reads 20.0%, one pooled standard error above flat. On Cube one hierarchical configuration leads flat by 3 to 5 points at every distance. Our flat CEM row at PushT t=75 (19.9%) matches their flat baseline, which places the full layer's 94.9% there on a comparable protocol.

M FURTHER RELATED WORK

DDIM (Song et al., 2020) and consistency samplers (Song & Dhariwal, 2024) cheapen single drafter calls, which compounds with the accept rule's call savings; TRM (Li et al., 2026a) and SCALE (Hu et al., 2026) repair latent metrics at training time, whereas we leave the metric frozen and calibrate a test on it. DSpark (Cheng et al., 2026b) schedules commitment with a learned confidence head, whereas our accept rule has no learned parameters; action-chunking critics meet the same trust question one level down, over *actions* (Liu et al., 2025; Black et al., 2026; Wu et al., 2026). ARC-Bench (Zhang & Li, 2026) shows that frequent closed-loop replanning masks ranking failures of frozen JEPA planners; the layer leaves the executor's replanning schedule untouched and economizes drafter calls only. Among further concurrent work on the same family of benchmarks, INTACT (Sun et al., 2026) amortizes goal-reaching into a search-free executor, IMWM (Gao et al., 2026) gates an

Table 23: **All configurations over LeFlow.** Success rate (%) by goal distance t, mean over eight evaluation seeds with its standard error. Vanilla LeFlow is shown at both of its horizons.

PushT	Configuration	$t = 25$	$t = 50$	$t = 75$	$t = 100$	$t = 150$
LeFlow	vanilla (twenty-five-step horizon)	99.6 ±0.1	83.3 ±0.5	49.8 ±0.6	31.4 ±0.6	5.6 ±0.4
	vanilla (ten-step horizon)	88.1 ±0.3	88.3 ±0.7	66.8 ±0.8	47.7 ±0.5	24.4 ±0.9
	drafter, every-step	61.9 ±0.6	26.6 ±0.4	20.5 ±0.7	14.9 ±0.5	8.9 ±0.4
	blind chain	78.0 ±0.7	65.8 ±0.8	68.0 ±1.2	66.0 ±0.8	68.9 ±0.3
	accept rule	61.8 ±0.5	26.3 ±0.7	19.9 ±0.5	14.8 ±0.8	11.2 ±0.3
	full (accept rule + arbiter)	**74.0** ±0.3	**26.9** ±0.5	**20.7** ±0.6	**15.0** ±0.6	**10.7** ±0.8

Reacher	Configuration	$t = 25$	$t = 50$	$t = 100$	$t = 150$
LeFlow	vanilla (twenty-five-step horizon)	88.3 ±1.0	98.6 ±0.2	91.1 ±0.4	80.6 ±0.5
	vanilla (ten-step horizon)	31.5 ±1.5	46.5 ±1.6	68.3 ±1.5	68.4 ±0.9
	drafter, every-step	19.4 ±0.9	31.2 ±1.0	50.1 ±1.0	60.6 ±1.2
	blind chain	20.9 ±1.4	27.1 ±1.6	41.0 ±0.9	55.0 ±1.0
	accept rule	22.3 ±1.3	33.0 ±1.3	49.1 ±1.9	57.7 ±1.4
	full (accept rule + arbiter)	**30.6** ±0.6	**40.4** ±1.1	**56.6** ±1.1	**63.7** ±0.6

Cube	Configuration	$t = 25$	$t = 50$	$t = 75$	$t = 100$	$t = 150$
LeFlow	vanilla (twenty-five-step horizon)	100.0 ±0.0	68.8 ±0.4	56.6 ±0.1	73.4 ±0.3	36.5 ±0.2
	vanilla (ten-step horizon)	66.7 ±0.6	58.4 ±0.5	38.0 ±0.7	20.3 ±0.5	38.5 ±0.5
	drafter, every-step	26.6 ±0.9	23.1 ±0.8	29.7 ±1.1	15.1 ±0.6	26.7 ±0.8
	blind chain	45.9 ±1.1	51.4 ±1.3	58.7 ±0.6	64.8 ±1.0	60.4 ±1.1
	accept rule	33.2 ±0.8	31.1 ±1.1	36.0 ±0.6	27.2 ±0.5	39.1 ±1.0
	full (accept rule + arbiter)	**64.9** ±0.8	**54.7** ±0.7	**41.0** ±0.8	**25.3** ±0.4	**39.5** ±0.9

Two-Room	Configuration	$t = 25$	$t = 50$	$t = 75$
LeFlow	vanilla (twenty-five-step horizon)	98.8 ±0.4	93.8 ±1.3	88.7 ±1.3
	vanilla (ten-step horizon)	85.9 ±1.2	94.1 ±1.1	96.3 ±0.7
	drafter, every-step	71.5 ±1.9	81.4 ±1.5	93.5 ±1.0
	blind chain	65.2 ±2.0	69.3 ±1.8	79.3 ±1.8
	accept rule	69.0 ±2.2	81.6 ±2.3	90.6 ±0.7
	full (accept rule + arbiter)	**76.8** ±1.5	**85.9** ±2.0	**94.1** ±1.3

action prior by an estimate of its reliability, which asks a cousin of the arbiter's question, EV-WM (Wang et al., 2026) and ProWorld (Liu et al., 2026) pair a world model with an imagination-based check of the plan, and VLWM (Du et al., 2026) evaluates Two-Room under a third protocol that we do not import, a sensitivity that Singh (2026) documents independently; LEAP (Pham & Bera, 2026) and Traj-LeWM (Huang et al., 2026b) are a further executor and a further cost over the same checkpoints.