When Best Prediction Fails: Adjacent Set Action Reconstruction for Latent World Model Planning

Anonymous authors
Paper under double-blind review

Abstract

Controllers based on sampling and latent world models rank candidate action sequences by predicted terminal cost, execute the first action block, and replan. Residual prediction error can give an infeasible sequence an anomalously low cost, even when the terminal objective reflects physical task progress. A larger proposal pool creates more opportunities for such errors to outrank feasible alternatives. We call this conditional failure *proposal overgeneration*. In Cube candidate execution audits, increasing the proposal budget from 72 to 288 reduces minimum latent cost selection feasibility from .375 to .062 for position targets and from .344 to .031 for position and yaw targets, although every larger pool contains a feasible sequence. We introduce Adjacent Set Action Reconstruction (ASAR). Among low-cost proposals, ASAR identifies an adjacent set using standardized early action prefixes, reconstructs a locally weighted action sequence with a light minimum-cost anchor, and executes its first block. On 75 Carry and Release queries, Kernel ASAR improves event completion success over minimum cost selection by 28.0, 24.0, and 18.7 percentage points under latent cost and by 18.7, 20.0, and 17.3 points under a trajectory reachability cost at 72, 144, and 288 proposals. Finite-pool analysis characterizes lower-tail selection risk, radius support under an explicit probability gap, and containment of the executed action block under a local physical condition.

1 Introduction

Control based on sampling and latent world models encodes the current and goal observations, predicts the terminal latent state for each candidate action sequence, and ranks the candidates with a terminal cost. A receding-horizon controller then executes the first action block of the selected sequence and replans. This pattern appears in latent model predictive control and recent world models for visual planning (Hafner et al., 2019; Hansen et al., 2022; 2024; Hafner et al., 2025; Micheli et al., 2023; Pu et al., 2025; Zhou et al., 2025; Assran et al., 2025; Maes et al., 2026b). Increasing the proposal budget improves candidate coverage, and a better terminal cost improves the evaluation of predicted endpoints. Neither operation explicitly uses distances between the action prefixes in the current proposal pool.

Figure 1 shows why these relations within the pool can matter. The controller begins with a grasped cube and must transport it to a lower target before release. The sequence with minimum latent cost has an isolated prefix in the full standardized prefix space. After its first action block is chosen at each replan, the cube finishes 21.2 cm from the target. Twelve other sequences with low cost have nearby prefixes. Kernel ASAR reconstructs a weighted full action sequence from that adjacent set and the sequence with minimum cost. The resulting closed-loop execution releases the cube within 3.8 cm of the target from the same initial state and matched proposal draws.

The failure begins because the controller minimizes costs computed from predicted terminal states, while execution reaches terminal states affected by residual model error. A useful predictor may still assign an infeasible sequence a cost below every feasible alternative. When more infeasible sequences are sampled from a comparable conditional distribution, the probability of observing at least one such error in the lower tail increases. Feasible sequence coverage can therefore improve

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.

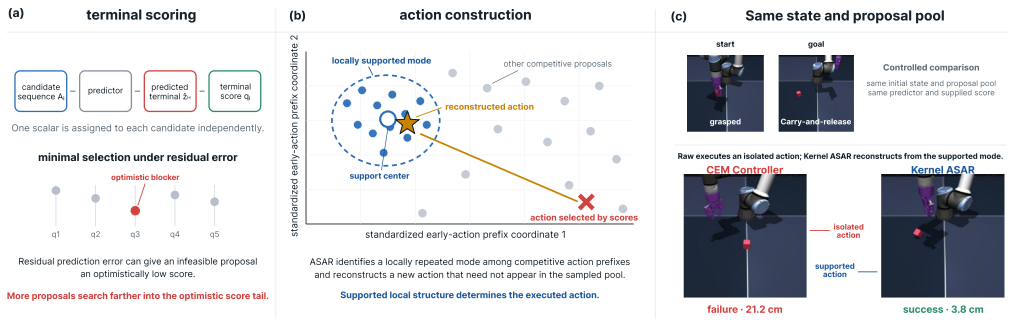


Figure 1: From terminal scoring to adjacent set action reconstruction. (a) A fixed predictor assigns a terminal score to each candidate independently, so residual prediction error can place an infeasible proposal in the optimistic score tail. (b) ASAR identifies a locally repeated mode among competitive early action prefixes and reconstructs a new full action sequence from this adjacent set. (c) In Carry and Release, the controllers share the initial state and pool, fixed predictor, terminal score, and matched proposal noise. Minimum score selection executes an isolated action and leaves the cube 21.2 cm from the target, while Kernel ASAR reconstructs from the adjacent set and completes the task within 3.8 cm. The full-sequence schematics represent an implementation whose first block is executed before replanning.

while the fraction of pools with a feasible sequence among the lowest costs decreases. We refer to this budget-induced degradation in candidate ranking under residual model error as proposal overgeneration.

We measure this pattern in simulated Cube manipulation with a fixed LEWM checkpoint and CEM controller. In 192 pools audited by independent candidate execution, minimum latent cost selects an infeasible sequence in 132 pools; 58 of these failures (43.9%) have a feasible alternative among the 80 lowest costs. Across proposal budgets, feasible presence remains high while top-20 exposure declines. A trajectory reachability cost improves ranking, and infeasible sequences with lower cost remain.

These results motivate an adjacent-set action reconstruction rule. When competitive first action blocks occupy a locally regular physical regime, nearby early prefixes provide a planning-time signal of repeated action structure. ASAR first restricts attention to sequences with low cost. It measures each sequence by the mean distance to its nearest standardized prefixes, identifies an adjacent set around the least isolated prefix, and reconstructs a weighted full action sequence from that set. A light mixture with the sequence of minimum cost retains information from the task objective. The world model, proposal generator, and terminal cost are unchanged. Selecting the least isolated existing sequence from the same set of sequences with low cost is the matched existing-candidate baseline without reconstruction.

Terminal objective quality and residual prediction error occur at different stages. Trajectory Reachability Metrics (TRM) replace terminal latent distance with a trajectory-based cost for predicted endpoints (Li et al., 2026a). ASAR addresses residual endpoint prediction error after the cost is chosen. We compare minimum cost selection and reconstruction under both latent distance and TRM.

Our contributions are:

1. We identify proposal overgeneration under residual model error: feasible coverage can increase while minimum cost selection becomes less reliable.

2. We propose ASAR, which reconstructs the next action block from adjacent low-cost prefixes through a weighted sequence that need not occur in the pool.

3. We give a lower-tail identity, a radius-support concentration result under an explicit probability gap, and a conditional containment bound for the executed block.

4. ASAR improves event completion under both costs across three budgets. A targeted test shows a 13.7-point gain over least isolated selection, and independent model weights replicate reconstruction gains.

2 Related Work

Latent world model control. PlaNet established online planning through learned latent dynamics, while Dreamer and DreamerV3 optimize behavior in imagined trajectories (Hafner et al., 2019; 2020; 2025). TD-MPC and TD-MPC2 combine compact prediction with receding-horizon control (Hansen et al., 2022; 2024). IRIS and UniZero study scalable planning in learned latent models (Micheli et al., 2023; Pu et al., 2025). Recent visual models plan over pretrained or jointly learned embeddings (Zhou et al., 2025; Assran et al., 2025; Maes et al., 2026b;a). Other work improves the planning interface or the model horizon (Parthasarathy et al., 2025; Du et al., 2026). We leave the encoder, predictor, and action sampler fixed and use only relations among sequences in the final proposal pool.

MPC and CEM. CEM iteratively samples candidates, retains an elite set, and refits its proposal distribution (Rubinstein, 1999). It is widely used with learned dynamics because it does not require differentiating through the environment (Chua et al., 2018; Hafner et al., 2019). iCEM improves sample efficiency through colored noise sampling, elite reuse, and distribution constraints (Pinneri et al., 2021), while MPPI forms action updates weighted by cost (Williams et al., 2017). DecentCEM maintains several independent search distributions to preserve multimodal structure (Zhang et al., 2022). Learned action priors, diffusion policies, and generators conditioned on subgoals also improve the distribution of candidate behavior (Chi et al., 2024; Wang et al., 2026; Cheng et al., 2026). These methods use population information in sampling or control updates. KDPE uses kernel density estimation to filter potentially harmful trajectories generated by Diffusion Policy (Rosasco et al., 2025). ASAR applies local aggregation to competitive prefixes in the final world-model proposal pool and executes the reconstructed first block.

Model error, objective quality, and reachability. Learned planners can select actions that exploit model error, and predictive accuracy under one loss need not imply accuracy for the planning objective (Janner et al., 2019; Lambert et al., 2020). Choosing the minimum among noisy estimates is also related to the optimizer's curse (Smith & Winkler, 2006). Reachability methods replace geometric proximity with evidence about controllable progress from replay graphs, latent landmarks, temporal distances, or directed quasimetrics (Eysenbach et al., 2019; Zhang et al., 2021; Qian et al., 2023; Bae et al., 2025; Myers et al., 2024; 2025). RC-aux was introduced to change world model training through multihorizon reachability supervision (Li et al., 2026b), while TRM was introduced to change the terminal objective for a predicted endpoint (Li et al., 2026a). ASAR addresses the subsequent residual failure. Given a meaningful terminal cost, it uses prefix distances within the pool to identify an adjacent set and reconstruct the sequence passed to receding-horizon execution.

3 Definitions and Control Pattern

Let $\mathbf{x}_t \in \mathcal{X}$ be the physical state at one replanning step and $o_t = g(\mathbf{x}_t)$ its observation. An encoder e maps the observation to $\mathbf{z}_t = e(o_t)$ and the goal observation o_g to $\mathbf{z}_g = e(o_g)$. A candidate *action sequence* is $\mathbf{A}_i = (\mathbf{a}_{i,0}, \ldots, \mathbf{a}_{i,H-1}) \in \mathcal{A}^H$, where one model action is an action block of repeated environment actions. The learned predictor produces

$$\hat{\mathbf{z}}_{i,H} = \hat{f}_H(\mathbf{z}_t, \mathbf{A}_i). \tag{1}$$

The controller assigns a terminal cost:

$$q_i = Q(\hat{\mathbf{z}}_{i,H}, \mathbf{z}_g). \tag{2}$$

Lower values are preferred. In the latent cost condition, $Q(\hat{\mathbf{z}}, \mathbf{z}_g) = \|\hat{\mathbf{z}} - \mathbf{z}_g\|_2^2$. In the reachability cost condition, Q is the trajectory-based cost from TRM (Li et al., 2026a). The two costs are inputs to the same reconstruction rule. In either case, q_i is computed from candidate i without using the executed outcome of another candidate.

For a candidate execution audit, let F_H denote the physical transition map for the full horizon. It reaches $\mathbf{x}_{i,H}^{\text{exec}} = F_H(\mathbf{x}_t, \mathbf{A}_i)$ and therefore

$$\mathbf{z}_{i,H}^{\text{exec}} = e\big(g(\mathbf{x}_{i,H}^{\text{exec}})\big)\,. \tag{3}$$

The predicted terminal latent $\hat{\mathbf{z}}_{i,H}$ and the executed terminal latent $\mathbf{z}_{i,H}^{\text{exec}}$ are different objects. The planner uses the former. The audit uses the latter only to label the candidate after an independent execution. In closed-loop control, the first action block is executed before a new pool is generated. Candidate-horizon feasibility and closed-loop task completion are separate outcomes. Once this block changes the physical state, the next replan starts from that new state; unexecuted alternatives from the previous state remain counterfactual.

Let $\ell_H(\mathbf{x}, o_g)$ include the terminal task conditions used by the audit, including target error and, when required, the release condition. For tolerance ϵ, the feasible set over the full horizon is

$$\mathcal{E}_\epsilon(\mathbf{x}_t, o_g) = \{\mathbf{A} \in \mathcal{A}^H : \ell_H(F_H(\mathbf{x}_t, \mathbf{A}), o_g) \le \epsilon\}. \tag{4}$$

For a proposal pool $\mathcal{C}_B = \{\mathbf{A}_i\}_{i=1}^B$, write $Y_i = 1$ when $\mathbf{A}_i \in \mathcal{E}_\epsilon$. A *ranking blocker* under Q is an infeasible proposal b with

$$q_b < \min_{i:Y_i=1} q_i. \tag{5}$$

The blocker count is the number of infeasible proposals with cost below that of the feasible proposal with the lowest cost. It is a diagnostic for selection by minimum cost, not a claim that every reconstruction rule is blocked by each such proposal.

In receding-horizon CEM, three fixed action noise scales produce a proposal pool. The world model predicts every candidate, and the elite set with the lowest costs updates the sampling distribution. After the last iteration, the controller executes the first action block of the selected sequence, observes the next image, and replans. The active terminal cost drives the elite update. We use *total proposal budget* for the number of sequences in one scored pool, separate from cumulative model calls. A *query* is one closed-loop evaluation from a start state to a goal state, which contains several replanning pools.

4 Proposal Scaling under Model Error

At total proposal budget B, let P_B be the event that at least one feasible sequence is present in the pool. Let $T_{B,k}$ be the event that at least one feasible sequence has rank at most k under q. The top-k feasibility event satisfies

$$\Pr(T_{B,k}) = \Pr(P_B)\Pr(T_{B,k} \mid P_B). \tag{6}$$

The first factor measures proposal coverage. The second measures whether a feasible sequence remains near the top after scoring. The identity is exact; the empirical question is how the two factors change with B, Q, and k.

We test this decomposition in a separate candidate execution evaluation set with 32 pools for each target family. Every candidate in each pool is executed independently from the same initial state to obtain Y_i. The two cost rules are then applied to the identical candidates. Table 1 and Figure 2 show that the reachability cost improves top-20 coverage at the larger budget, while infeasible sequences with lower cost remain.

The same candidate execution audit also measures top-1 feasibility. From 72 to 288 total proposals, the feasibility of the selected sequence falls from .375 to .062 for position targets and from .344 to .031 for targets defined by position and yaw under latent cost. Under the reachability cost, the corresponding rates are .281 to .188 and .281 to .156. These budgets characterize the evaluated proposal procedure, whose adaptive distributions can also change with budget.

For a fixed state, model, objective, proposal distribution, and threshold t, condition on n_- infeasible proposals. Let $F_-(t) = \Pr(q_b \le t \mid Y_b = 0)$ under this conditioning. If these costs are independent with common distribution F_-,

$$\Pr\left(\min_{b:Y_b=0} q_b \le t\right) = 1 - (1 - F_-(t))^{n_-}\,. \tag{7}$$

Table 1: Candidate execution audit of proposal overgeneration. Each target family has 32 pools and every candidate is executed independently. At both total budgets, every pool contains at least one feasible sequence. Top-20 exposure is the fraction of pools with a feasible sequence among the 20 lowest costs. Blockers is the mean number of infeasible sequences with cost below that of the feasible sequence with the lowest cost.

		Top-20 exposure		Mean blockers	
Target family	Cost	$B = 72$	$B = 288$	$B = 72$	$B = 288$
Position	Latent	.969	.656	4.31	16.91
Position	Reachability	.875	**.750**	6.66	**16.66**
Position + yaw	Latent	.969	.625	4.34	18.31
Position + yaw	Reachability	.906	**.719**	7.59	**17.75**

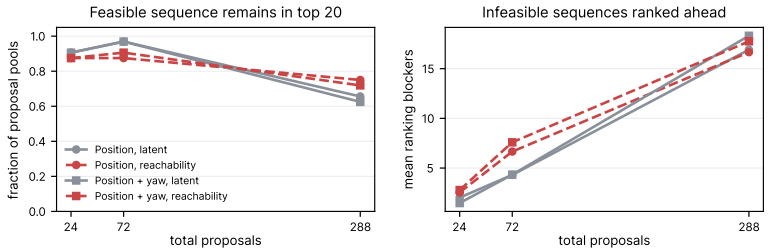


Figure 2: Proposal scaling in the candidate execution audit. The fraction of pools with a feasible sequence among the 20 candidates with the lowest costs declines from 72 to 288 total proposals, although feasible presence is already one at both budgets. Mean counts of ranking blockers increase in the same comparison. The reachability cost changes the level of the curves but does not remove the effect of the larger pool.

For fixed $0 < F_{-}(t) < 1$, this probability increases strictly with n_{-}. The identity isolates sample count at a fixed distribution. The budget audit measures the combined effect of count and the evaluated proposal procedure. The appendix gives the corresponding blocker expectation with a random feasible threshold.

The candidate execution audit connects this ranking pattern to physical outcomes. In 58 of 132 failures under latent cost, a feasible sequence is already among the 80 candidates with the lowest costs. In a separate diagnostic with 48 pools, replacing predicted terminal latents with latents encoded after independent execution raises minimum cost selection from .292 to .500, while ranking by the true terminal physical loss reaches .542. These are diagnostic upper bounds for selection over one horizon, not information available to the controller. Full accounting and the control with shuffled reachability costs appear in the supplement.

5 Adjacent Set Action Reconstruction

5.1 Adjacent Set Identification

The terminal cost evaluates candidates separately. ASAR adds one relation that is available only after the current pool has been generated: whether a prefix with low cost has nearby prefixes from other candidates. The intended setting is a locally regular action region, where small changes in a prefix are likely to preserve the same behavior over a short horizon.

Let $\mathcal{R}_M(Q)$ be the M candidates with lowest terminal cost. For each candidate, vectorize the first h model action blocks into an action prefix $\mathbf{p}_i$ and robustly standardize each coordinate inside $\mathcal{R}_M$. Define the local prefix isolation score as the mean distance to the K nearest other prefixes,

$$u_i = \frac{1}{K} \sum_{j \in \mathcal{N}_K(i)} \|\mathbf{p}_i - \mathbf{p}_j\|_2. \tag{8}$$

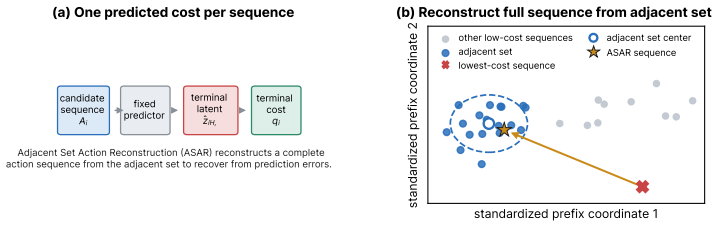


Figure 3: Adjacent Set Action Reconstruction. **(a)** A fixed predictor maps each candidate action sequence to one terminal prediction and task cost. The cost may be latent distance or a trajectory reachability cost. **(b)** Among candidates with low cost, ASAR uses standardized prefix distances to identify an adjacent set. It reconstructs a new full action sequence by weighted averaging and adds a 10% anchor from the sequence with minimum cost. The plotted coordinates are a two-dimensional visualization of the prefix features, not the full action space.

The neighborhood excludes i. Smaller u_i indicates greater local prefix density within the competitive pool. This proxy uses no executed outcomes.

The proxy has a related finite-sample interpretation. Fix two query prefixes e, b, their feature transform, radius r, and an independent reference sample of n prefixes. Let $S_r(i)$ be the fraction of reference prefixes within r of i. If $\mathbb{E}S_r(e) \geq p_+ > p_- \geq \mathbb{E}S_r(b)$, independent reference draws and Hoeffding's inequality (Hoeffding, 1963) give

$$\Pr(S_r(e) \leq S_r(b)) \leq 2\exp\left[-\frac{n(p_+ - p_-)^2}{2}\right]. \tag{9}$$

This is a sufficient condition for radius support under independent reference sampling. The implemented KNN statistic uses the same pool for standardization and selection. Its empirical diagnostic (Appendix L) finds a positive isolation gap in .548 of 31 pairs for the best-scored feasible candidate ($p = .3605$), and .774 for the least isolated feasible candidate ($p = .0032$). This second pairing selects for local support. These observations measure association; the local physical condition below is an additional assumption.

5.2 Action Reconstruction and Execution

Kernel ASAR uses this proxy to choose the candidate c with smallest u_i, identifies an adjacent set $\mathcal{S}$ around its prefix, and assigns

$$w_i \propto \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_c\|_2^2}{d_p \tau_d} - \lambda \tilde{q}_i\right), \quad i \in \mathcal{S}, \tag{10}$$

where d_p is the prefix dimension, $\tilde{q}_i$ is the robustly standardized terminal cost, and weights sum to one. Let $\mathbf{A}_Q = \arg\min_i q_i$ be the sequence with minimum cost in the eligible pool. The reconstruction operator produces a full action sequence

$$\mathbf{A}_{\text{rec}} = (1-\alpha)\sum_{i \in \mathcal{S}} w_i \mathbf{A}_i + \alpha \mathbf{A}_Q. \tag{11}$$

The first term aggregates the local set; the second retains the minimum-cost sequence. The action parameterization must permit coordinatewise interpolation. Writing $\mathbf{b}_i = \mathbf{a}_{i,0}$, the executed block is exactly $\mathbf{b}_{\text{rec}} = (1-\alpha)\sum_i w_i \mathbf{b}_i + \alpha \mathbf{b}_Q$. Later reconstructed blocks are discarded at replanning. Figure 3 shows the operation in a two-dimensional visualization of prefix space.

The executed block has a conditional containment property. Suppose every block in the adjacent set satisfies $\|\mathbf{b}_i - \mathbf{b}_c\| \leq R_0$ and let $D_0 = \|\mathbf{b}_Q - \mathbf{b}_c\|$. Convexity gives

$$\|\mathbf{b}_{\text{rec}} - \mathbf{b}_c\| \leq (1-\alpha)R_0 + \alpha D_0. \tag{12}$$

If the ball of radius γ_0 around $\mathbf{b}_c$ lies in a physically admissible set of first blocks, a right-hand side at most γ_0 keeps the reconstruction in that set. This condition concerns the next execution step. The appendix gives the proof and its full-sequence counterpart. Re-scoring the reconstruction through the same world model measures model consistency; it supplies no independent physical observation to establish this condition.

6 EXPERIMENTS

We test proposal scaling, reconstruction beyond existing-candidate selection, replication across budgets and model weights, and the limits of transfer.

Environment and controller. We use simulated Cube manipulation from OGBench (Park et al., 2025) with a fixed pretrained LEWM encoder, predictor, and CEM generator. The generator draws equally from action-noise scales 1.0, 1.5, and 2.0. Total proposal budgets are 72, 144, and 288. The controller has 40 environment steps. Each model action represents a block of five environment actions, and the controller replans after each block. Kernel ASAR retains $M = \min(80, B)$ lowest-cost sequences, computes u_i from the first model action with $K = \min(3, M - 1)$, takes the 12 nearest sequences around the least isolated center within the 32 least isolated candidates, and uses $\alpha = .10$. These parameters are fixed across reported evaluations. At evaluation time, no method observes the executed outcomes of candidates in the current pool.

Matched comparison of terminal costs. The Carry and Release evaluation set contains 75 queries defined by start and goal states from 44 evaluation seeds. Some seeds contribute more than one query, so confidence intervals resample seeds as clusters. Query membership is defined from the associated expert trajectory before planner outcomes are inspected. Contact is present at the start and absent at the goal, and the cube is lowered by more than 3 cm. We compare two terminal costs, latent distance and TRM reachability, with three ways to produce the next sequence. Minimum cost selection and selection of the least isolated sequence each return an existing candidate. Kernel ASAR reconstructs a new full action sequence. All six combinations use matched random proposal noise. Identical states and sampling distributions yield identical pools; state divergence and cost-dependent elite updates can change later pools. The reachability head uses 100,000 trajectory pairs and excludes every episode represented in this evaluation set or the candidate execution scaling audit.

Outcomes and additional evaluation sets. The primary outcome for Carry and Release is event completion success. It requires the environment success indicator and a final contact state consistent with release. Other Cube evaluations report the environment success indicator. A rescue is a query where ASAR succeeds and its stated baseline fails; a loss is the reverse. Confidence intervals are 95% bootstrap intervals clustered by seed from 10,000 resamples.

A separate targeted test contains 73 queries from 54 new evaluation seeds. Its predefined rule uses translation between the start and goal, yaw change, and the maximum kernel weight in the initial pool. The rule selects cases where the reconstruction weight is spread across several adjacent candidates. This set is used to compare reconstruction with selection of the least isolated existing candidate, not to estimate performance over the full query distribution. The canonical Cube, final-pool control, and independent-model evaluations each contain 512 queries and are distinct panels. Exact selection rules, episode exclusions, hyperparameter provenance, transfer tests, and all secondary controls appear in the supplement.

7 RESULTS

7.1 ADJACENT-SET RECONSTRUCTION IMPROVES UNDER BOTH TERMINAL COSTS

Table 2 and Figure 4 compare two terminal costs and three output rules. Under latent cost, Kernel ASAR improves event completion success over minimum cost selection by 28.0 percentage points at 72 proposals, 24.0 points at 144, and 18.7 points at 288. The corresponding 95% intervals are $[16.0, 40.0]$, $[14.3, 34.2]$, and $[6.8, 31.2]$ points. Using the reachability cost gives gains of 18.7, 20.0, and 17.3 points with intervals $[9.6, 28.0]$, $[10.7, 30.1]$, and $[6.8, 27.1]$.

The baseline that selects the least isolated sequence controls for using the same prefix-density proxy without reconstructing a new sequence. Under reachability cost, Kernel ASAR exceeds this baseline by 14.7 points at 72 proposals with interval $[3.9, 25.0]$, and by 10.7 points at 144 with interval $[1.3, 21.1]$. At 288, the gain is 9.3 points with interval $[-3.0, 20.3]$. Thus the first two budgets give direct evidence for reconstruction beyond selection of an existing candidate, while the larger-budget comparison remains uncertain. The reachability cost changes the terminal ranking of predicted

Table 2: Event completion success on the same 75 Carry and Release queries from 44 evaluation seeds. Columns compare selection of an existing candidate with Kernel ASAR under two terminal costs. All methods use matched proposal draws.

Total	Latent cost			Reachability cost		
proposals	Minimum	Least isolated	Kernel ASAR	Minimum	Least isolated	Kernel ASAR
72	.067	.187	**.347**	.147	.187	**.333**
144	.053	.213	**.293**	.133	.227	**.333**
288	.133	.147	**.320**	.133	.213	**.307**

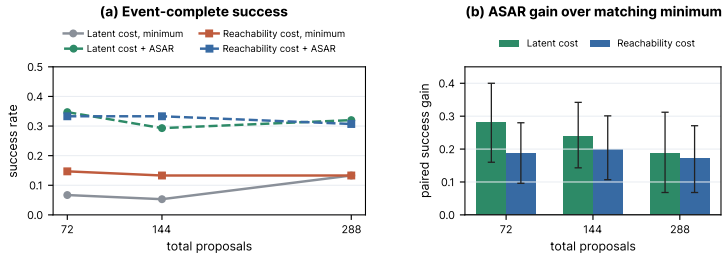


Figure 4: Carry and Release results across total proposal budgets. (A) Event completion success for minimum cost selection and Kernel ASAR under latent and reachability costs. (B) Paired Kernel ASAR gain over the matching baseline based on minimum cost. Error bars are 95% bootstrap intervals that resample the 44 evaluation seeds.

endpoints, while ASAR changes the executed action after scoring. Gains under both costs show that the two interventions act at distinct stages of the planning pipeline.

7.2 Reconstruction Improves beyond Selection of an Existing Candidate

The targeted 73-query test compares two rules built from the same 80 sequences with low cost and the same prefix distances. Selecting the least isolated existing sequence reaches .493 environment success. Reconstructing a full action sequence with Kernel ASAR reaches .630, a gain of 13.7 percentage points with 95% interval $[1.9, 23.8]$. The corresponding mean Cube error, target position error (m), and yaw error (rad) change from .154, .071, and .843 to .135, .047, and .762. The appendix defines these recorded fields. Table 3 reports this comparison beside the aggregate and evaluation with an independent model with each outcome and baseline stated explicitly.

With independently trained world model weights, the least isolated existing sequence reaches .717 and Kernel ASAR reaches .738 at 72 proposals. Their paired difference is 2.1 points with interval $[1.0, 3.5]$. This directly replicates the reconstruction advantage without relying on the original model weights. Figure 1 illustrates the corresponding physical execution.

7.3 Where Adjacent-Set Reconstruction Helps

On the 75 Carry and Release queries at 288 proposals, Kernel ASAR rescues 19 queries that fail under selection by minimum latent cost and loses 5 queries that the baseline solves. The remaining 51 outcomes are unchanged. A separate 512-query evaluation gives success under selection by minimum latent cost .711 and Kernel ASAR success .742, a paired gain of 3.1 points with interval $[1.0, 5.5]$. On the 42 queries whose expert segment contains grasp, lift, transport, lower, and release within one interval from start to goal, event completion success under selection by minimum latent cost is .119 and Kernel ASAR success is .048. A single adjacent prefix set does not supply the several distinct behaviors missing from these longer queries.

Ordinary elite averaging, the arithmetic mean of the five lowest-cost sequences, reaches .721 on the same final-pool control set (Appendix Table 15). It is an output-rule control for averaging without

Table 3: Closed-loop comparisons across Cube evaluation sets. Each row states its outcome and baseline, so rates and paired differences are interpreted only within that row. The independent-model row uses 72 total proposals; other rows use 288. The Carry and Release row is from the separate budget sweep in Appendix Table 19.

Evaluation set	Outcome	N	Baseline rule	Baseline	Kernel ASAR	Δ [95% CI]
Cube aggregate	Environment	512	Latent minimum	.723	.762	+.039 [+.016,+.065]
Independent model	Environment	512	Latent minimum	.703	.738	+.035 [+.014,+.057]
Targeted reconstruction	Environment	73	Least isolated	.493	.630	+.137 [+.019,+.238]
Carry and Release	Event complete	75	Latent minimum	.133	.320	+.187 [+.069,+.310]

adjacency. Kernel ASAR is 2.15 points higher with interval $[0.0, 4.3]$, and its continuous errors also improve with intervals above zero. The supplement tests other explanations using terminal state density, action index penalties, filtering based only on the graph, and random filters matched to the graph retention rate. None matches the Kernel point estimate on that evaluation set.

Portfolio ASAR is also evaluated on two different domains. On Reacher, its success is .796 compared with .800 for the standard CEM mean, with interval clustered by seed $[-.064, .044]$. On Push-T, success under latent cost changes from .273 to .280 with interval $[-.080, .100]$; under a true state hybrid cost it changes from .413 to .393. These results show that the Cube prefix metric is not a universal action-space metric. The positive Cube result appears when feasible sequences with low cost occupy a locally dense prefix region that is suitable for interpolation.

8 Discussion and Limitations

Increasing a proposal budget changes both coverage and selection risk. Reporting feasible presence, top-k feasible coverage, and ranking-blocker counts separates these effects. TRM changes the terminal cost (Li et al., 2026a); ASAR reconstructs after scoring, allowing both interventions in one controller.

Three conditions characterize the scope of ASAR: a useful low-cost candidate set, feasible candidates with greater local support than ranking blockers, and reconstructed first blocks inside a locally admissible physical neighborhood. Dense infeasible clusters can attract reconstruction; isolated feasible candidates can be rejected; missing stages of a long behavior cannot be supplied by local averaging. Medoids, graph filtering, and resettable controller mixtures retain their respective physical assumptions.

The positive closed-loop evidence uses two independently trained models in simulated Cube. Carry and Release begins with an already grasped cube. Transfer is neutral on Reacher and neutral or negative on Push-T. The radius support result assumes a probability gap, and block containment additionally assumes an admissible ball. The candidate audit measures support ordering; the frequency of physically admissible balls remains unmeasured. Hardware evaluation should examine sensing noise, failure recovery, and safe reset costs.

9 Conclusion

Residual prediction errors can make minimum cost selection less reliable as proposal pools grow. ASAR reconstructs the next block from adjacent low-cost prefixes. Candidate audits expose the ranking failure, matched control evaluations establish gains under two terminal costs, and targeted tests isolate reconstruction from existing-candidate selection.

AI USE STATEMENT

Generative AI tools assisted literature search, code and workflow preparation, manuscript editing, and LaTeX preparation. The authors are responsible for verifying AI-assisted content and for the final claims, experiments, code, and artifacts.

REPRODUCIBILITY STATEMENT

The paper reports the model, planning, candidate-audit, and paired-evaluation protocols needed to interpret the main results. Additional implementation details, finite-pool analyses, complete Cube results, transfer evaluations, and environment information are provided in the appendix.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Junik Bae, Kwanyoung Park, and Youngwoon Lee. TLDR: Unsupervised goal-conditioned rl via temporal distance-aware representations. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2183–2204. PMLR, 2025. URL https://proceedings.mlr.press/v270/bae25a.html.

Letian Cheng, Qi Zhang, and Yisen Wang. SAGE: Subgoal-conditioned action generation for latent world model planning. *arXiv preprint arXiv:2607.17973*, 2026. URL https://arxiv.org/abs/2607.17973.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024. doi: 10.1177/02783649241273668. URL https://doi.org/10.1177/02783649241273668.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems*, 31, 2018. URL https://arxiv.org/abs/1805.12114.

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. URL https://arxiv.org/abs/2606.21775.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1906.05253.

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

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.01603.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2. URL https://doi.org/10.1038/s41586-025-08744-2.

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

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 761–770. PMLR, 2020. URL https://proceedings.mlr.press/v120/lambert20a.html.

Liangyu Li, Shengzhi Wang, and Qingwen Liu. Beyond euclidean proximity: Repairing latent world models with horizon-matched trajectory reachability metrics. *arXiv preprint arXiv:2605.22164v1*, 2026a. URL https://arxiv.org/abs/2605.22164v1.

Wenyuan Li, Guang Li, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Predictive but not plannable: RC-aux for latent world models. *arXiv preprint arXiv:2605.07278*, 2026b. URL https://arxiv.org/abs/2605.07278.

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. URL https://arxiv.org/abs/2605.21800.

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

Vincent Micheli, Eloi Alonso, and Fran
ccois Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37076–37096. PMLR, 2024. URL https://proceedings.mlr.press/v235/myers24a.html.

Vivek Myers, Bill Zheng, Benjamin Eysenbach, and Sergey Levine. Offline goal-conditioned reinforcement learning with quasimetric representations. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0661. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1c5956164472d6d8123d574aa75cd063-Abstract-Conference.html.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ecd92623ac899357312aaa8915853699-Abstract-Conference.html.

Arjun Parthasarathy, Nimit Kalra, Rohun Agrawal, Yann LeCun, Oumayma Bounou, Pavel Izmailov, and Micah Goldblum. Closing the train-test gap in world models for gradient-based planning. *arXiv preprint arXiv:2512.09929*, 2025. URL https://arxiv.org/abs/2512.09929.

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

Yuan Pu, Yazhe Niu, Zhenjie Yang, Jiyuan Ren, Hongsheng Li, and Yu Liu. UniZero: Generalized and efficient planning with scalable latent world models. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=Gl6dF9soQo.

Zhifeng Qian, Mingyu You, Hongjun Zhou, Xuanhui Xu, and Bin He. Goal-conditioned reinforcement learning with disentanglement-based reachability planning. *arXiv preprint arXiv:2307.10846*, 2023. URL https://arxiv.org/abs/2307.10846.

Andrea Rosasco, Federico Ceola, Giulia Pasquale, and Lorenzo Natale. KDPE: A kernel density estimation strategy for diffusion policy trajectory selection. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 1210–1224. PMLR, 2025. URL https://proceedings.mlr.press/v305/rosasco25a.html.

Reuven Y. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999.

James E. Smith and Robert L. Winkler. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science*, 52(3):311–322, 2006. doi: 10.1287/mnsc.1050.0451.

Yuhai Wang, Jiawei Xia, Rongxuan Zhou, Xiao Hu, Yongliang Shi, Jing Du, and Yang Ye. PRISM: PRior-guided imagination sampling in world models. *arXiv preprint arXiv:2606.07974*, 2026. URL https://arxiv.org/abs/2606.07974.

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

Lunjun Zhang, Ge Yang, and Bradly C. Stadie. World model as a graph: Learning latent landmarks for planning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12611–12620. PMLR, 2021. URL https://proceedings.mlr.press/v139/zhang21x.html.

Zichen Zhang, Jun Jin, Martin Jagersand, Jun Luo, and Dale Schuurmans. A simple decentralized cross-entropy method. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2212.08235.

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

A Appendix Overview

This supplement expands the diagnosis, method, theory, and experimental protocol in the main paper. It first defines the physical execution setting and specifies each ASAR reconstruction variant. It then states the finite proposal pool results, records the Cube protocol, and reports candidate execution analyses and matched closed-loop comparisons. Later sections provide replication with independently trained world model weights, joint latent and action neighborhood controls, a targeted reconstruction evaluation, a bounded mixture of controllers, and transfer results on Reacher and Push-T.

Cube is the main method setting because its low cost proposal pools contain both ranking blockers and locally dense action prefixes. Reacher and Push-T test the unchanged Cube reconstruction rule when this prefix geometry differs. The theory mirrors this scope. The lower tail identity concerns costs in a finite pool. The radius support result requires an explicit local probability gap. The reconstruction bound requires closeness and a robust physical neighborhood in the space being executed; its first-block corollary applies to the Cube controller. Physical value is measured through paired closed-loop execution.

B Notation and Physical Execution

Let $\mathbf{x}_t \in \mathcal{X}$ be the physical state at one replanning step and let $o_t = g(\mathbf{x}_t)$ be the corresponding observation. The encoder e produces $\mathbf{z}_t = e(o_t)$ and the goal latent $\mathbf{z}_g = e(o_g)$. A candidate action sequence is

$$\mathbf{A}_i = (\mathbf{a}_{i,0}, \dots, \mathbf{a}_{i,H-1}) \in \mathcal{A}^H. \tag{13}$$

One model action $\mathbf{a}_{i,h}$ is an action block that represents five environment actions in Cube. An action prefix contains the first h model actions of the full sequence. The learned predictor maps the current latent and action sequence to a predicted terminal latent,

$$\hat{\mathbf{z}}_{i,H} = \hat{f}_H(\mathbf{z}_t, \mathbf{A}_i). \tag{14}$$

In a candidate execution audit, F_H denotes counterfactual physical execution of the full sequence. The terminal state and its encoded observation are

$$\mathbf{x}_{i,H}^{\text{exec}} = F_H(\mathbf{x}_t, \mathbf{A}_i), \qquad \mathbf{z}_{i,H}^{\text{exec}} = e\big(g(\mathbf{x}_{i,H}^{\text{exec}})\big). \tag{15}$$

Equations 14 and 15 name different objects. The planner ranks $\hat{\mathbf{z}}_{i,H}$. The audit obtains $\mathbf{z}_{i,H}^{\text{exec}}$ only after executing candidate i from the same stored initial state. Closed-loop control does not execute F_H for every candidate. It executes only the first action block of the selected sequence, observes the next image, and replans.

A terminal cost maps the predicted endpoint and goal to one scalar,

$$q_i = Q(\hat{\mathbf{z}}_{i,H}, \mathbf{z}_g). \tag{16}$$

Lower values are preferred. Latent distance and TRM reachability are the two evaluated instances of Q. All subsequent prefix operations condition on this pool ordered by terminal cost.

Let ℓ_H include the full horizon terminal conditions used by an audit, including target error and release state when required. For tolerance ϵ, the feasible sequence set is

$$\mathcal{E}_\epsilon(\mathbf{x}_t, o_g) = \{\mathbf{A} : \ell_H(F_H(\mathbf{x}_t, \mathbf{A}), o_g) \le \epsilon\}. \tag{17}$$

Inside a finite proposal pool $\mathcal{C}_B$, let $Y_i = 1$ indicate $\mathbf{A}_i \in \mathcal{E}_\epsilon$. A ranking blocker under Q is an infeasible proposal b whose cost is lower than every feasible cost,

$$Y_b = 0, \qquad q_b < \min_{i:Y_i=1} q_i. \tag{18}$$

The total proposal budget B counts action sequences in one replanning pool. A query denotes one closed-loop evaluation from a start state to a goal state and may contain several replanning pools.

Table 4: Fixed Kernel ASAR parameters used in the confirmatory runs.

Item	Value
Eligible low cost candidates	80
Prefix steps	1
KNN neighbors	3
Prefix density candidate pool	32
Adjacent set size	12
Kernel temperature τ	0.75
Terminal cost energy weight	0.35
Minimum cost anchor α	0.10

C Detailed ASAR Implementations

C.1 Candidate Pool and Terminal Cost

At each replan, the fixed world model receives the current image, the goal image, and a shared pool of action sequences $\{\mathbf{A}_i\}_{i=1}^{N}$, where $\mathbf{A}_i \in \mathbb{R}^{H \times d_a}$. It predicts terminal latent states $\hat{\mathbf{z}}_i$. The terminal cost rule assigns

$$q_i = Q(\hat{\mathbf{z}}_i, \mathbf{z}_g). \tag{19}$$

For latent cost, $q_i = \|\hat{\mathbf{z}}_i - \mathbf{z}_g\|_2^2$. For reachability cost, Q is the learned temporal head described in Section F. All output rules compared under one cost use the same initial state, goal, fixed world model, proposal generator, matched proposal noise, and execution budget.

For the Cube evaluations, the proposal generator produces 96 sequences at each of three variance scales $1.0, 1.5, 2.0$, for 288 candidates per replan. The ASAR reconstruction variants retain $M = \min(80, B)$ candidates with lowest terminal cost. The listed 80-candidate setting is the cap for all reported budgets. This restriction preserves goal competitiveness while excluding obviously poor proposals.

C.2 Local Prefix Isolation Score

Let $\mathbf{p}_i$ be the vectorized first model action of candidate i. For Cube this prefix contains one action block, corresponding to five environment actions. Each prefix coordinate is robustly standardized within the current eligible pool using the median and interquartile range, falling back to the standard deviation when the interquartile range is degenerate.

For each eligible candidate, the implementation computes the mean Euclidean distance to its $k = \min(3, M-1)$ nearest other standardized prefixes. If $M = 1$, it returns the sole candidate. For $M > 1$,

$$u_i = \frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} \|\mathbf{p}_i - \mathbf{p}_j\|_2. \tag{20}$$

Smaller distance means that a prefix is less isolated in the eligible pool. The distances are robustly standardized again before use. Sorting by u_i defines the prefix density order. The least isolated baseline returns the existing candidate at the head of this order. The two main ASAR variants instead reconstruct a full action sequence from an adjacent set. This statistic is available at planning time and serves as a density proxy, but it does not by itself establish physical feasibility.

C.3 Kernel ASAR Sequence Reconstruction

Kernel ASAR is implemented with the fixed parameters summarized in Table 4. The terminal cost term in the kernel weight uses the robustly standardized cost defined for the current proposal pool.

The least isolated candidate supplies the prefix center. Among the first 32 candidates in prefix density order, the method identifies the 12 prefixes nearest to this center as the adjacent set. Let d_p be the prefix feature dimension, let $\bar{d}_i^2 = \|\mathbf{p}_i - \mathbf{p}_c\|_2^2 / d_p$, and let $\tilde{q}_i$ denote the robustly standardized

terminal cost. The unnormalized weight is

$$\tilde{w}_i = \exp\left[-\left(\frac{\bar{d}_i^2}{0.75} + 0.35\tilde{q}_i\right)\right]. \tag{21}$$

After normalization, the sequence reconstructed from the adjacent set is

$$\mathbf{A}_{\mathrm{ker}} = \sum_{i \in \mathcal{S}_{12}} w_i \mathbf{A}_i. \tag{22}$$

The output retains the sequence with minimum cost $\mathbf{A}_Q$:

$$\mathbf{A}_{\mathrm{out}} = 0.90\mathbf{A}_{\mathrm{ker}} + 0.10\mathbf{A}_Q. \tag{23}$$

No executed outcome, success label, or physical state obtained from the simulator enters the reconstruction operator.

C.4 Portfolio ASAR Sequence Reconstruction

Portfolio ASAR identifies the seven least isolated candidates as an adjacent set $\mathcal{E}$. In robustly standardized prefix space, define

$$\rho = \frac{\|\mathbf{p}_Q - \bar{\mathbf{p}}_{\mathcal{E}}\|_2}{\frac{1}{|\mathcal{E}|}\sum_{i \in \mathcal{E}} \|\mathbf{p}_i - \bar{\mathbf{p}}_{\mathcal{E}}\|_2 + \epsilon}. \tag{24}$$

The fixed branch rule is:

- **Close,** $\rho \leq 1.0$**:** use the elite arithmetic mean and anchor from the sequence with minimum cost $\alpha = 0.25$.
- **Middle,** $1.0 < \rho < 1.75$**:** use the elite medoid in prefix space and anchor from the sequence with minimum cost $\alpha = 0.15$.
- **Far,** $\rho \geq 1.75$**:** use the geometric median of the full elite action sequences and anchor from the sequence with minimum cost $\alpha = 0.10$.

For representative $\mathbf{A}_{\mathrm{rep}}$, the output is always

$$\mathbf{A}_{\mathrm{out}} = (1-\alpha)\mathbf{A}_{\mathrm{rep}} + \alpha\mathbf{A}_Q. \tag{25}$$

The medoid representative is an original sampled candidate; the subsequent minimum-cost anchor can produce a new sequence. The mean and geometric median representatives aggregate the adjacent set. Proposition 5 states the additional local conditions under which the reconstructed sequence remains inside the feasible set.

C.5 Joint Latent and Action Neighborhood Reconstruction

The joint neighborhood extension first filters the 80 lowest cost candidates before applying the same Portfolio ASAR rule. For each candidate, robustly standardized predicted terminal latent features are concatenated with the robustly standardized action prefix containing one model action. The joint features are reduced to at most 16 principal components. An undirected graph is then formed by connecting each candidate to its eight nearest neighbors.

The five nodes with the smallest mean neighbor distance define the dense neighborhood seeds. Shortest path distance from these seeds is computed for all eligible candidates. The method retains the lower 50% of these distances, subject to a minimum of 16 and a maximum of 48 candidates. With 80 eligible candidates, the retained neighborhood contains 40 candidates. The Portfolio ASAR prefix density order, elite set, branch ratio, representative, and minimum cost anchor are recomputed inside this retained neighborhood. The anchor is the candidate with minimum cost inside the retained neighborhood, not the global cost minimum.

The random control matched to the retention rate uses the same retained size and is calibrated to match the joint neighborhood method's empirical rate of filtering out the candidate with minimum cost. It therefore distinguishes generic pruning from filtering based on the joint latent and action neighborhood.

C.6 Optional Bounded Mixture of Controllers

The Mixture of Controllers (MoC) is a supplementary bounded verification extension, separate from the single execution ASAR method. It treats uncertainty across reconstruction variants as a bounded verification problem. For each query it executes a fixed controller order: Portfolio ASAR, Kernel ASAR, anchored adjacent prefix mean, and adjacent prefix mean. If an earlier branch succeeds within the 40-step Cube budget, evaluation stops. If it fails, the next branch is run from the same initial state and goal. Thus, the protocol allocates a bounded branch budget across controller families and evaluates them sequentially, stopping at the first successful branch.

The four concrete output rules are:

1. Portfolio ASAR;
2. Kernel ASAR;
3. the adjacent prefix mean with a 0.25 anchor on the sequence with minimum cost;
4. the unanchored adjacent prefix mean.

The verifier never inspects later outcomes before deciding to continue. In the current protocol, failed branches reset to the same initial state. This matches repeatable simulation and resettable control settings. Irreversible deployment requires a safe state recovery mechanism in place of reset.

C.7 Planning Procedure

At each replan, ASAR performs the following operations using information available at planning time:

1. Sample candidates at the fixed proposal scales and roll them through the fixed world model.
2. Compute the terminal cost and retain its $\min(80, B)$ lowest-cost candidates.
3. Extract the first action block, robustly standardize the prefixes, and compute the $k = 3$ nearest-neighbor prefix isolation score.
4. For the kernel variant, restrict to the 32 least isolated candidates, take the nearest 12, and apply Eqs. 21–23. For the portfolio, take seven adjacent prefix candidates and apply Eqs. 24–25. The graph extension first filters to the retained joint latent and action neighborhood and then applies the same portfolio rule.
5. Execute the first five environment actions, observe the next image, and repeat until success or the 40-step Cube budget is exhausted.

All operations use only the current proposal pool, predicted latent rollouts, terminal costs, and action sequences.

MoC wraps four complete instances of this planner. It evaluates them in the fixed order given above, stops on the environment success signal, and resets to the same stored initial state only after a failed branch. The controller order and branch limit do not depend on outcomes from later controllers.

D Finite Proposal Pool Results and Proofs

The controller chooses from a finite proposal pool at each replan. The statements below analyze that decision object directly using proposal costs, adjacent prefix sets, and reconstructed action sequences. This pool level view also accommodates contact transitions where the physical map is not globally continuous.

D.1 Growth of Lower Tail Ranking Errors

Proposition 1 (risk from lower tail overgeneration). Fix the state, model, terminal cost, proposal distribution, and threshold t. Condition on the infeasible proposal labels and their count

n_-. Suppose their terminal costs are conditionally independent with common lower-tail distribution $F_-(t) = \Pr(q_b \le t \mid Y_b = 0)$. Then

$$\Pr\left(\min_{b:Y_b=0} q_b \le t\right) = 1 - (1 - F_-(t))^{n_-}. \tag{26}$$

For any threshold with $0 < F_-(t) < 1$, the probability is strictly increasing in n_-.

Proof. The event $\min_b q_b > t$ requires every infeasible proposal to have score greater than t. Conditional independence gives $\Pr(\min_b q_b > t) = (1 - F_-(t))^{n_-}$. Taking the complement proves Eq. 26. Its one step increase is $F_-(t)(1 - F_-(t))^{n_-} > 0$. □

For a pool containing a feasible proposal, let $T_B = \min_{i:Y_i=1} q_i$ and $\mathcal{I}_- = \{j : Y_j = 0\}$. The blocker count is $M_B = \sum_{j\in\mathcal{I}_-} \mathbf{1}[q_j < T_B]$. Without an independence assumption, linearity of expectation gives

$$\mathbb{E}[M_B \mid T_B, \mathcal{I}_-] = \sum_{j\in\mathcal{I}_-} \Pr(q_j < T_B \mid T_B, \mathcal{I}_-). \tag{27}$$

Each probability conditions on the random best feasible score and the label set. Under equal conditional tails, the sum is $|\mathcal{I}_-|$ times that common conditional probability. Its dependence on B, including changes caused by adaptive sampling, is retained. Proposition 1 concerns a fixed threshold and fixed distribution; the empirical budget sweep can change both sample count and proposal distribution.

Proposition 2 (presence and top-k retention). At budget B, let P_B be the event that the proposal pool contains at least one feasible sequence. Let $T_{B,k}$ be the event that at least one feasible sequence appears among the k lowest cost proposals. Since $T_{B,k} \subseteq P_B$,

$$\Pr(T_{B,k}) = \Pr(P_B)\Pr(T_{B,k} \mid P_B). \tag{28}$$

Proof. The set identity $T_{B,k} = T_{B,k} \cap P_B$ and the multiplication rule give Eq. 28. □

The two factors distinguish candidate generation from retention based on cost. When $\Pr(P_B)$ is already high, a reduction in $\Pr(T_{B,k} \mid P_B)$ can lower the probability that a feasible sequence is available to a top-k output rule.

D.2 Concentration of Local Prefix Support

Fix a feature transform, a radius r, and query prefixes e, b before drawing a reference sample $\{\mathbf{r}_j\}_{j=1}^n$ independently of those queries and the transform. Define support by

$$S_r(i) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}[\rho(\mathbf{p}_i, \mathbf{r}_j) \le r]. \tag{29}$$

Proposition 3 (local support separation). Let e be feasible and b a ranking blocker. Condition on the queries, their labels, and the fixed transform. Suppose reference draws are independent, so radius-membership indicators are independent across j for each query, and satisfy

$$\mathbb{E}[S_r(e)] \ge p_+, \qquad \mathbb{E}[S_r(b)] \le p_-, \qquad p_+ > p_-. \tag{30}$$

Then

$$\Pr(S_r(e) \le S_r(b)) \le 2\exp\left[-\frac{n(p_+ - p_-)^2}{2}\right]. \tag{31}$$

Proof. Let $m = (p_+ + p_-)/2$ and $\Delta = p_+ - p_-$. If $S_r(e) \le S_r(b)$, then either $S_r(e) \le m$ or $S_r(b) \ge m$. Hoeffding's inequality gives

$$\Pr(S_r(e) \le m) \le \exp(-n\Delta^2/2), \tag{32}$$

$$\Pr(S_r(b) \ge m) \le \exp(-n\Delta^2/2). \tag{33}$$

The union bound proves Eq. 31. □

The two support sums may share the same reference sample; the proof uses a union bound and requires no independence between those sums. The implemented statistic in Eq. 20 uses KNN distances, same-pool standardization, and same-pool candidate selection. These adaptive operations are outside the independent-reference condition of Proposition 3. Both summarize local prefix density, with opposite numerical directions: larger S_r and smaller u_i indicate a less isolated prefix. The candidate execution audit in Section L evaluates the exact KNN statistic used by the controller.

D.3 Existing Candidate Reranking by Prefix Isolation

Proposition 4 (condition for pairwise reranking). Let e be a feasible candidate and b a ranking blocker with $q_b < q_e$. Let u_i be the observable prefix isolation score from Eq. 20, and define $s_i = q_i + \beta u_i$ with $\beta > 0$. Then

$$u_b - u_e > \frac{q_e - q_b}{\beta} \tag{34}$$

if and only if $s_e < s_b$.

Proof. Multiplying Eq. 34 by β and rearranging gives $q_e + \beta u_e < q_b + \beta u_b$. This is exactly $s_e < s_b$. Reversing the algebra proves the converse. □

Proposition 4 describes a rule for reranking existing candidates. The audit in Section L measures its premise with executed candidates. Kernel ASAR and Portfolio ASAR use the same local geometry to reconstruct a new full action sequence, whose outcome is measured through paired closed-loop execution.

D.4 Containment of Reconstructed Full Sequences

Proposition 5 (containment of a reconstructed sequence). Let $\mathcal{S}$ be an adjacent set identified by prefix geometry, and let $\mathbf{A}_c$ be a reference full sequence. Suppose $\|\mathbf{A}_i - \mathbf{A}_c\| \le R$ for every $i \in \mathcal{S}$, the weights satisfy $w_i \ge 0$ and $\sum_i w_i = 1$, the anchor satisfies $0 \le \alpha \le 1$, and the sequence with minimum cost obeys $\|\mathbf{A}_Q - \mathbf{A}_c\| = D$. For

$$\mathbf{A}_{\text{out}} = (1-\alpha)\sum_{i\in\mathcal{S}} w_i \mathbf{A}_i + \alpha \mathbf{A}_Q, \tag{35}$$

we have

$$\|\mathbf{A}_{\text{out}} - \mathbf{A}_c\| \le (1-\alpha)R + \alpha D. \tag{36}$$

If the closed ball $\{\mathbf{A} : \|\mathbf{A} - \mathbf{A}_c\| \le \gamma\}$ lies inside $\mathcal{E}_\epsilon$ and the right-hand side of Eq. 36 is at most γ, then $\mathbf{A}_{\text{out}} \in \mathcal{E}_\epsilon$.

Proof. The triangle inequality and convexity of the norm give

$$\begin{aligned}
\|\mathbf{A}_{\text{out}} - \mathbf{A}_c\| &\le (1-\alpha)\left\|\sum_i w_i(\mathbf{A}_i - \mathbf{A}_c)\right\| \\
&\quad + \alpha\|\mathbf{A}_Q - \mathbf{A}_c\| && (37)\\
&\le (1-\alpha)\sum_i w_i\|\mathbf{A}_i - \mathbf{A}_c\| + \alpha D && (38)\\
&\le (1-\alpha)R + \alpha D. && (39)
\end{aligned}$$

The final statement follows from containment in the robust feasible ball. □

Proposition 5 formalizes the two reconstruction terms. A compact reconstruction set reduces R, while the anchor from the sequence with minimum cost contributes according to its displacement D. Here R, D, and γ are measured in the space of complete action sequences. Prefix proximity selects the reconstruction set but does not by itself imply this full sequence bound. The bound therefore states the additional local condition under which weighted reconstruction remains inside a robust feasible neighborhood.

Corollary (containment of the executed first block). Let Π_0 project a sequence onto its first action block and write $\mathbf{b}_i = \Pi_0 \mathbf{A}_i$. Linearity gives

$$\mathbf{b}_{\mathrm{rec}} = \Pi_0 \mathbf{A}_{\mathrm{out}} = (1-\alpha) \sum_i w_i \mathbf{b}_i + \alpha \mathbf{b}_Q. \tag{40}$$

If $\|\mathbf{b}_i - \mathbf{b}_c\| \le R_0$ and $D_0 = \|\mathbf{b}_Q - \mathbf{b}_c\|$, the same triangle-inequality proof gives

$$\|\mathbf{b}_{\mathrm{rec}} - \mathbf{b}_c\| \le (1-\alpha) R_0 + \alpha D_0. \tag{41}$$

To state a physical consequence, fix a next-state criterion ψ_0 and tolerance ϵ_0, and define $\mathcal{U}_0 = \{\mathbf{b} : \psi_0(G_5(\mathbf{x}_t, \mathbf{b})) \le \epsilon_0\}$, where G_5 executes the five primitive actions in one Cube block. If the ball of radius γ_0 around $\mathbf{b}_c$ lies in $\mathcal{U}_0$ and the bound is at most γ_0, the reconstructed block lies in $\mathcal{U}_0$. This is a sufficient condition on the next block, separate from full-task success under subsequent replanning. The reported support statistics do not measure containment of an entire physical ball.

All radii in this corollary use the same norm on raw action coordinates. With a fixed diagonal scale matrix S and standardization $\mathbf{p} = S^{-1}(\mathbf{b} - \boldsymbol{\mu})$, a standardized displacement of norm at most r implies raw displacement at most $\|S\|_{\mathrm{op}} r$ in the Euclidean norm. This conversion addresses action geometry; physical admissibility is the additional ball-containment assumption. The implementation discards the reconstructed suffix at the next replan.

D.5 Separation of Proposal Neighborhoods

Proposition 6 (separation of proposal clusters). Suppose a proposal pool is partitioned into clusters and every pair from different clusters is separated by more than δ in the graph metric. If graph edges connect only pairs at distance at most $\epsilon < \delta$, then no connected component contains proposals from two clusters.

Proof. An edge between different clusters would require a pair from different clusters at distance at most ϵ, contradicting the assumed separation. Any path that crosses clusters must contain such an edge. □

The medoid of a component is one of its sampled proposals. Selecting that medoid therefore introduces no interpolation beyond the original candidate set. Proposition 5 gives the corresponding local condition for a weighted mean with a score anchor. Returning the unanchored medoid preserves membership in the sampled set; adding the anchor again requires the interpolation condition. Proposition 6 concerns separation in the observed proposal graph and does not assign a physical regime label to a component.

D.6 Optional Bounded Mixture of Controllers

Proposition 7 (success union and expected branch count). Let S_j be the success event for controller j from the shared initial query, and let MoC execute controllers in a fixed order, stopping at the first verified success. Then

$$\Pr(\text{MoC succeeds}) = \Pr\left(\bigcup_{j=1}^{J} S_j \right). \tag{42}$$

If L is the number of executed branches, then

$$\mathbb{E}[L] = \sum_{j=1}^{J} \Pr\left(\bigcap_{\ell<j} S_\ell^c \right). \tag{43}$$

Proof. The verifier returns success exactly when at least one branch succeeds, which proves Eq. 42. Branch j is executed exactly when all earlier branches fail. Since $L = \sum_{j=1}^{J} \mathbf{1}[L \ge j]$, linearity of expectation gives Eq. 43. □

The result requires no independence among controller outcomes. It states the benefit and branch cost measured by validation from the same initial state on 768 new queries.

Table 5: Cube closed-loop evaluation protocol.

Item	Fixed setting
Environment and data	`ogbench/cube_single_expert` (Park et al., 2025); fixed pretrained `cube/lewm` encoder and dynamics.
Queries	8 paired queries per seed; identical start and goal for every method within a query.
Goal offset and budget	Goal offset 40 environment steps; evaluation budget 40 environment steps.
Model horizon	8 model actions, each representing an action block of 5 environment actions.
Replanning	Execute 5 environment actions, then replan; at most 8 replans.
Proposal pool	96 candidates at each variance scale $1.0, 1.5, 2.0$, giving 288 candidates per replan.
Eligible sequences	80 lowest terminal costs; prefix length 1; $k = 3$ nearest-neighbor isolation score.
Execution	Every reconstructed sequence is executed in the same environment loop as the minimum cost sequence. No oracle that executes every candidate is used by the method.
Statistics	Paired success wins/losses; primary intervals resample evaluation seeds as clusters with 10,000 bootstrap repetitions. Intervals paired by query are retained as sensitivity analyses. Continuous improvements are baseline minus method so positive is better.

E Cube Evaluation Protocol

This section fixes the matched closed-loop protocol used for every Cube method comparison and records the disjoint evaluation splits and reproducibility information. ASAR receives only quantities visible to the planner; executed candidate outcomes are reserved for evaluation and the separate diagnostic audits.

E.1 Closed-Loop Evaluation Configuration

Table 5 specifies the environment, controller, execution budget, and statistical protocol.

E.2 Data Splits and Evaluation Roles

Table 6 distinguishes development, held-out candidate execution, paired control, and transfer evaluations.

E.3 Method Development and Confirmatory Evaluation Protocol

The eligible set size, prefix length, neighbor count, kernel parameters, and Portfolio ASAR branch settings were selected on development blocks before the canonical Cube evaluations. The joint neighborhood rule was developed on one query block and then applied without modification to a separate control block. The seven final proposal pool output rules were recorded before their evaluation.

The independent model replication fixes the independently trained Cube checkpoint, proposal budgets, output rules, and inclusion rule before the replication evaluation. The Cube reconstruction rule is unchanged. The MoC controller order was recorded after the router analysis and before its separate evaluation. Its primary comparison is MoC success against Portfolio first execution under repeated attempts from the same initial state.

Table 7 summarizes the comparisons used to select the final reconstruction variants. The anchor study evaluated weights $0, .10, .25, .35, .50$. It identified a small anchor weight on the sequence with minimum cost as the useful regime and showed that weights .35 and .50 attenuate the prefix density effect. The remaining parameters were fixed before the confirmatory evaluations.

Table 6: Experimental roles and split boundaries. Development, confirmatory, replication, and transfer evaluations are kept separate as specified below.

Experiment family	Scale	Role	Separation rule
Proposal overgeneration	12 development blocks, 4 held-out blocks, and 4 replication blocks	Diagnosis and replication	Held-out and replication blocks are not used for development
Canonical Cube evaluation	Two sets of 256 queries	Confirmatory method evaluation	Evaluation seeds are disjoint from method development
Final proposal pool controls	512 queries	Output rule controls	Seven output rules are fixed before execution
Joint neighborhood study	256 development and 256 control queries	Extension and matched control	The control evaluation is separate from development
Prefix density analysis	96 calibration and 96 held-out pools	Analysis of the candidate level premise	The calibration coefficient is fixed before the held-out analysis
Independent model replication	Two sets of 512 queries	Replication across model weights	Uses an independently trained Cube world model
Transfer evaluations	250 Reacher episodes and 150 paired Push-T episodes for each terminal cost	Boundary analysis	Uses the Cube reconstruction rule without parameter changes
Mixture of Controllers	768 queries	Supplementary evaluation with resets	Controller order is fixed before evaluation

Table 7: Development choices and settings used without change in the confirmatory evaluations.

Component	Development comparison	Confirmatory setting
Eligible set and local geometry	Competitiveness under minimum cost, prefix density, terminal density, and full sequence controls	80 lowest costs, prefix 1, $k = 3$
Minimum cost anchor	$0, .10, .25, .35, .50$ on development pools	.10 for Kernel and .25/.15/.10 by Portfolio branch
Portfolio representative	Local means, medoids, robust centers, and elite size variants	Seven elites with ratio thresholds 1.0 and 1.75
Kernel ASAR	Local kernel centers, adjacent set sizes, terminal cost weights, and anchors	Pool 32, set 12, $\tau = .75$, cost weight .35, anchor .10
Graph neighborhood	Latent, action, and joint graphs with random controls matched to the filtering rate	PCA 16, graph $k = 8$, five core seeds, median core distance retention

F Matched Evaluation of Terminal Costs and Output Rules

Relation to the trajectory reachability cost. Li et al. introduced Trajectory Reachability Metrics (TRM) to test whether terminal latent distance reflects physical task progress when scoring a predicted endpoint (Li et al., 2026a). The learned metric changes the terminal cost assigned to existing candidates. ASAR operates after this scoring step: it uses the geometry of early action prefixes in the current proposal pool and may reconstruct a full sequence that was not sampled. The matched comparison therefore crosses latent distance and TRM reachability with three output rules while holding the world model, proposal generator, evaluation queries, and proposal draws fixed.

Evaluation set. The set contains 75 Carry and Release queries grouped into 44 evaluation seeds. Query membership was recorded before the matched comparison using the same rule, defined independently of outcomes, as in the main paper. The associated expert segment begins in contact, ends without contact, and lowers the cube by more than 3 cm. The query definition file records the dataset row, episode, start and goal steps, evaluation seed, and query identifier for every case.

Episode exclusion. The temporal head is trained on the public `ogbench/cube_single_expert` trajectories (Park et al., 2025) after excluding 102 episodes. This union contains all 73 episodes represented by the Carry and Release evaluation set and all 32 episodes in the independent candidate execution set. Three episodes occur in both sets, yielding 102 unique excluded episodes in total.

Temporally Supervised Heads and Heads with Shuffled Labels. Both heads use the same pairwise architecture, 100,000 training pairs, 10,000 validation pairs, temporal range 1 to 40, target scale 40, optimizer schedule, and pair split. The temporal head learns within episode time separation. The control permutes only the training labels. The best validation RMSE is 8.97 for temporal supervision and 11.62 for shuffled supervision. Head training encodes 125,785 unique dataset rows and uses no evaluation episode.

Six matched controllers. For each terminal cost, latent distance or TRM reachability, the controller uses one of three output rules. Selection by minimum cost returns the existing candidate with minimum q_i. Selection of the least isolated candidate returns the existing candidate with the smallest prefix isolation score among the $\min(80, B)$ lowest-cost candidates. Kernel ASAR uses the same eligible set and prefix geometry, then reconstructs the anchored full sequence in Eq. 23. Proposal budgets are 24, 48, and 96 per source, or 72, 144, and 288 in total. The six controllers use matched evaluation seeds and proposal noise in one batched environment call.

Analysis. Event completion success requires the environment success indicator and final contact error at most 0.1. Contrasts are paired at the query level. Confidence intervals resample the 44 seeds as clusters with replacement for 10,000 bootstrap draws. The main contrasts compare Kernel ASAR with minimum cost selection and with least isolated existing sequence selection under each terminal cost. The difference between reconstruction gains estimates the interaction between terminal cost and output rule. The shuffled head run is evaluated at the largest proposal budget.

G Complete Proposal Overgeneration Results

G.1 Candidate Execution Diagnosis

The first audit executes candidates from 192 fixed proposal pools. It records whether the latent cost minimum succeeds and whether any successful candidate is already present among the 80 lowest latent costs. The second diagnostic uses 48 separate proposal pools and compares selection under predicted terminal latent distance, executed terminal latent distance, and executed terminal task loss.

Among the 132 pools where selection by minimum latent cost fails, 58 contain a successful candidate among the 80 lowest latent costs. This is 43.9% of failures under selection by minimum latent cost. In these pools the candidate has been generated, but predicted latent cost ranks it behind a failed candidate. The separate 48-pool result shows that this order changes substantially when the terminal latent is encoded after execution.

Table 8: Candidate execution diagnosis. The 192-pool accounting and 48-pool terminal state comparison use separate fixed candidate sets.

192-pool accounting	Pools	Rate
Latent minimum succeeds	60	.313
Latent minimum fails, top-80 success exists	58	.302
Latent minimum fails, no top-80 success	74	.385
48-pool selection cost	**Success**	**Δ predicted**
Predicted terminal latent	.292	–
Executed terminal latent	.500	+.208
Physical terminal loss	.542	+.250

G.2 Replication across Proposal Budgets

Table 9 reports the independent replication. The position and yaw targets reach complete feasible presence at 24 candidates per source, while top-20 feasible coverage falls sharply at 96. The cube state target behaves differently because its useful candidates appear later and across a broader range of proposal scales.

Table 9: Independent overgeneration replication on held-out proposal pools. Blockers are infeasible candidates ranked below the lowest cost of any feasible candidate.

Target	Budget/source	Present	Top-20	Top-1	Blockers
Position	8	.906	.906	.469	2.03
Position	24	1.000	.969	.375	4.31
Position	96	1.000	.656	.062	16.91
Position + yaw	8	.906	.906	.500	1.48
Position + yaw	24	1.000	.969	.344	4.34
Position + yaw	96	1.000	.625	.031	18.31
Cube	8	.625	.562	.062	7.70
Cube	24	.969	.656	.062	15.97
Cube	96	1.000	.562	.094	29.81

Table 10: Replication of the non-monotone top-20 feasible coverage pattern. Feasible presence is already one at the budget with highest top-20 coverage.

Split	Target	Peak budget	Peak top-20	Full top-20	Full blockers
Held out A	Position	24	.875	.500	23.66
Held out A	Position + yaw	24	.844	.594	20.38
Replication B	Position	24	.969	.656	16.91
Replication B	Position + yaw	24	.969	.625	18.31

G.3 Residual Lower Tail Errors under TRM

The independent candidate execution set is rescored with the temporal head and the matched shuffled label head. Feasibility labels continue to come only from executing the fixed candidates. Table 11 reports the two target families used in the main paper. Temporal TRM retains more useful candidates in the top 20 at the largest budget than latent cost or the shuffled head, but its blocker count still grows by roughly ten per pool. The executed top-1 rates from 72 to 288 total proposals fall from .375 to .062 and from .344 to .031 under latent cost. Under the reachability cost, they fall from .281 to .188 and from .281 to .156.

Table 11: Proposal scaling under latent, reachability, and shuffled label reachability costs. Column headings give total proposals across three sources. Each row contains 32 independently executed pools.

		Top-20 exposure			Mean blockers		
Target	Cost	24	72	288	24	72	288
Position	Latent	.906	.969	.656	2.03	4.31	16.91
Position	Reachability	.875	.875	.750	2.55	6.66	16.66
Position	Shuffled reach.	.875	.844	.625	3.97	9.16	22.03
Position + yaw	Latent	.906	.969	.625	1.48	4.34	18.31
Position + yaw	Reachability	.875	.906	.719	2.79	7.59	17.75
Position + yaw	Shuffled reach.	.875	.875	.594	4.34	6.94	23.41

H Complete Closed-Loop Cube Results

H.1 Matched Evaluation of Terminal Costs and Output Rules

Table 12 reports all six controllers. Kernel ASAR has the highest event completion success under both terminal costs at every tested budget. Reachability cost improves minimum cost selection at 72 and 144 proposals, while the two rules based on minimum cost are equal at 288.

Table 12: Event completion success for the matched terminal cost and output rule comparison. Each budget uses the same 75 queries and 44 evaluation seeds.

Total proposals	Output rule	Latent cost	Reachability cost
72	Minimum cost	.067	.147
72	Least isolated	.187	.187
72	Kernel ASAR	**.347**	**.333**
144	Minimum cost	.053	.133
144	Least isolated	.213	.227
144	Kernel ASAR	**.293**	**.333**
288	Minimum cost	.133	.133
288	Least isolated	.147	.213
288	Kernel ASAR	**.320**	**.307**

Relative to reachability selection by minimum cost, Kernel records 15/1, 16/1, and 16/3 paired rescues/losses as the proposal budget grows. Relative to the reachability least isolated rule, the counts are 15/4, 10/2, and 14/7. The terminal cost by Kernel interaction is $-.093$ $[-.189, .000]$, $-.040$ $[-.146, .068]$, and $-.013$ $[-.179, .139]$. We therefore report the paired Kernel gain under each cost directly without inferring a superadditive interaction.

The matched shuffled label head provides a second control at 288 proposals. Its event completion success under minimum cost selection is .187 and Kernel reaches .240, a paired gain of $+.053$ $[-.053, +.167]$ with 11 rescues and 7 losses. Temporal TRM at the same budget gives a Kernel

Table 13: Paired event completion success contrasts with 95% bootstrap intervals clustered by seed.

Total	Contrast	Δ [95% CI]
72	Reachability minimum minus latent minimum	+.080 [.000,.156]
72	Kernel ASAR with latent cost minus matching minimum	+.280 [.160,.400]
72	Kernel ASAR with reachability cost minus matching minimum	+.187 [.096,.280]
72	Kernel ASAR with latent cost minus matching least isolated	+.160 [.054,.264]
72	Kernel ASAR with reachability cost minus matching least isolated	+.147 [.039,.250]
144	Reachability minimum minus latent minimum	+.080 [.000,.156]
144	Kernel ASAR with latent cost minus matching minimum	+.240 [.143,.342]
144	Kernel ASAR with reachability cost minus matching minimum	+.200 [.107,.301]
144	Kernel ASAR with latent cost minus matching least isolated	+.080 [-.043,.200]
144	Kernel ASAR with reachability cost minus matching least isolated	+.107 [.013,.211]
288	Reachability minimum minus latent minimum	+.000 [-.099,.100]
288	Kernel ASAR with latent cost minus matching minimum	+.187 [.068,.312]
288	Kernel ASAR with reachability cost minus matching minimum	+.173 [.068,.271]
288	Kernel ASAR with latent cost minus matching least isolated	+.173 [.066,.290]
288	Kernel ASAR with reachability cost minus matching least isolated	+.093 [-.030,.203]

gain of $+.173$ $[+.068, +.271]$. The shuffled head does not reproduce the positive interval obtained with temporal supervision.

Table 14: Cube evaluation over 64 seeds and 512 paired queries. Wins and losses compare each method with selection by minimum latent cost.

Method	Success	Wins/Losses	Cube error	Replans
Latent minimum	.723	–	.1172	2.95
Least isolated	.744	17/6	.1161	2.82
Kernel ASAR	.762	26/6	.1061	2.70
Portfolio ASAR	**.766**	**27/5**	.1064	**2.69**

The success intervals clustered by seed relative to selection by minimum latent cost are [.0039,.0391] for least isolated selection, [.0156,.0645] for Kernel ASAR, and [.0215,.0664] for Portfolio ASAR. Kernel ASAR and Portfolio ASAR also have positive cube position and target position intervals. The same clustered bootstrap procedure is used for the reported intervals.

Kernel ASAR and Portfolio ASAR are close but not identical. Kernel ASAR succeeds on 26 queries that fail under selection by minimum latent cost and fails on 6 queries that this baseline solves. Portfolio ASAR succeeds on 27 queries that fail under selection by minimum latent cost and fails on 5 queries that this baseline solves. In direct comparison with Kernel ASAR, Portfolio ASAR succeeds alone on 10 queries and fails alone on 8. Their success union is .781. Held-out scalar routers did not reliably realize this union, so the paper reports the fixed reconstruction variants.

H.2 Controls for Reconstruction from the Final Proposal Pool

This evaluation fixes the model, proposal generation recipe, matched proposal noise, execution budget, and seven output rules over 64 new seeds and 512 paired queries. It is distinct from the canonical 512-query evaluation in Table 14. It tests whether adjacent-set reconstruction improves beyond generic averaging of low cost candidates.

Table 15: Final proposal pool reconstruction controls on 512 paired queries. Continuous improvement is the error under selection by minimum latent cost minus the method error, so positive is better.

Method	Success	Δ minimum	W/L minimum	Cube error improvement
Latent minimum	.7109	–	–	–
Latent elite mean	.7207	+.0098	15/10	.0054
Elite mean with minimum anchor	.7168	+.0059	11/8	.0067
Adjacent prefix mean	.7324	+.0215	21/10	.0078
Anchored adjacent prefix mean	.7344	+.0234	21/9	.0070
Kernel ASAR	**.7422**	**+.0312**	**24/8**	**.0116**
Portfolio ASAR	.7383	+.0273	24/10	.0098

Table 16: Kernel ASAR paired differences with 95% intervals clustered by seed. The success interval versus elite mean under latent cost touches zero. Both continuous error intervals are positive.

Comparison	Success	Cube error	Target pos.
vs. latent minimum	.0312 [.0098,.0547]	.0116 [.0069,.0167]	.0106 [.0061,.0151]
vs. latent elite mean	.0215 [.0000,.0430]	.0063 [.0019,.0106]	.0032 [.0003,.0062]
vs. anchored elite mean	.0254 [.0059,.0449]	.0049 [.0010,.0089]	.0044 [.0013,.0074]

Latent cost elite averaging gains roughly one point, while the Kernel ASAR gain over selection by minimum latent cost is 3.1 points. Kernel ASAR also leads elite averaging under latent cost by 2.15 points, with an interval whose lower endpoint is zero, and improves both continuous errors with positive intervals. We use Kernel ASAR as the main reconstruction variant and Portfolio ASAR as a robustness variant.

H.3 Carry and Release Phase Analysis

The second 512-query evaluation also supports an analysis by physical task phase. The subset rules were recorded before the outcomes were inspected and use only the expert segment associated with each start and goal. The main outcome is event completion success. It requires environment success and a final gripper contact value within .1 of the goal contact value.

Kernel ASAR improves Carry and Release event success by .187 over selection by minimum latent cost with an interval of [+.069,+.310] obtained by resampling seeds and 19 rescues against 5 losses. Its environment success gain over elite mean under latent cost is .133 with interval [+.025,+.247]. The broader release transition shows the same direction. Full pick and place queries still require the controller to compose grasp, lift, transport, lower, and release within one start and goal interval. Kernel ASAR reduces target position error there but does not improve event completion success.

The deterministic cases in Figure 5 provide the physical interpretation of the final proposal pool aggregate. They begin after grasp formation, so the controller must preserve contact during transport and then match a released goal. The long horizon boundary is reported separately through the full pick and place row in Table 18.

Table 17: Phase definitions specified independently of outcomes. Carry and Release begins from an already grasped cube and ends at a released goal.

Query subset	Expert segment rule
Release transition	Start contact above .5 and goal contact below .1
Carry and Release	Release transition with cube height decreasing by more than 3 cm
Full pick and place	Start and goal height below 4 cm, maximum height above 7 cm, contact above .5, and XY displacement above 3 cm

Table 18: Outcomes by physical task phase on the second 512-query evaluation. Target position error is final object to goal distance and lower is better.

Query subset	Method	(N)	Event success	Environment success	Target position error
Release transition	Latent minimum	91	.154	.198	.186
	Latent elite mean	91	.242	.253	.138
	Kernel ASAR	91	**.308**	**.352**	**.128**
	Portfolio ASAR	91	.297	.341	.134
Carry and Release	Latent minimum	75	.133	.147	.191
	Latent elite mean	75	.213	.213	.137
	Kernel ASAR	75	**.320**	**.347**	**.125**
	Portfolio ASAR	75	**.320**	.333	.134
Full pick and place	Latent minimum	42	**.119**	**.119**	.184
	Latent elite mean	42	.071	.071	.187
	Kernel ASAR	42	.048	.071	**.177**
	Portfolio ASAR	42	.024	.071	.178

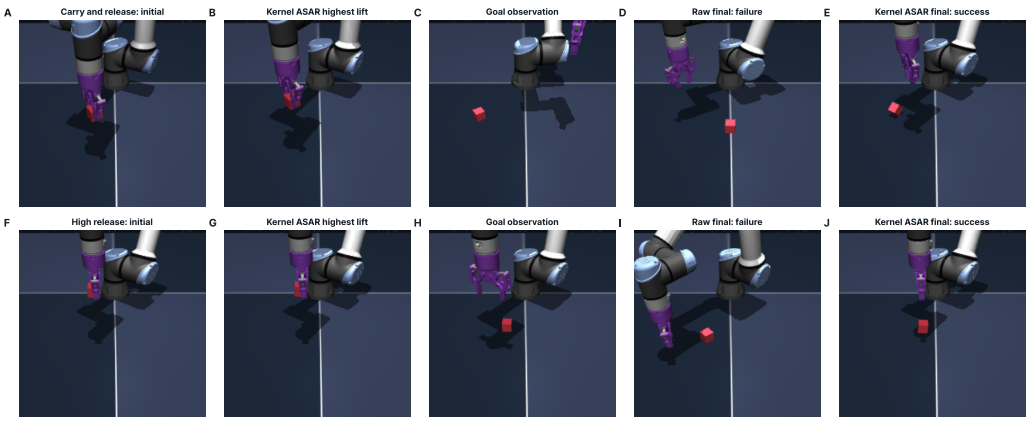


Figure 5: Carry and Release executions from the matched paired evaluation. (A-E) A grasped cube is transported to a lower goal. Kernel ASAR lifts the cube, places it near the target, and releases it, while selection by minimum latent cost releases far from the goal. (F-J) A second high release case shows the same physical transition. Every panel is a simulator observation from the executed trajectory.

Table 19: Event completion success in matched Carry and Release evaluations across proposal budgets. Total proposal counts are 72, 144, and 288.

Budget/source	Latent minimum	Latent elite	Kernel ASAR	Portfolio ASAR
24	.067	.147	**.360**	.347
48	.053	.253	.293	**.347**
96	.133	.213	**.347**	.307

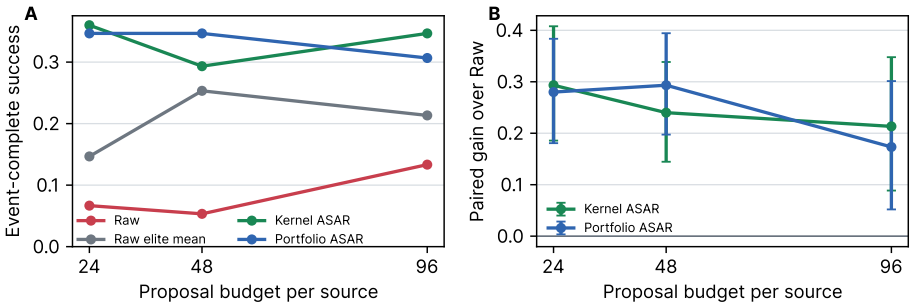


Figure 6: Carry and Release performance across proposal budgets. (A) Fixed controller outcomes. (B) Paired gains over selection by minimum latent cost with 95% intervals clustered by seed. Kernel ASAR improves that baseline by .293, .240, and .213 as the total proposal count increases from 72 to 288.

H.4 Carry and Release Results across Proposal Budgets

This budget sweep and the terminal-cost factorial comparison in Table 12 are separate recorded runs. Their original outcomes and intervals are reported within each panel; rates from the two runs are not combined.

We rerun the same 75 initial states with 24, 48, and 96 proposals per source. All evaluations share the 201,000-row dataset, model checkpoints, metadata for the evaluation queries, controller definitions, and 40-step budget. The row audit matches seed, query identifier, dataset row, episode, start step, and goal step. Pools are regenerated per budget, so the sweep measures closed-loop performance across independently generated pools at each budget.

Kernel ASAR improves selection by minimum latent cost at every budget with intervals [+.186,+.408], [+.145,+.338], and [+.089,+.348]. The selection by minimum latent cost improves from budget 48 to 96 by .080 with interval [+.013,+.153], so this Carry and Release set does not reproduce the degradation in the candidate execution audit. The position and yaw candidate sets provide the direct proposal overgeneration evidence.

I Cube Results Stratified by Task Conditions

We separate static and moving targets to locate the aggregate Cube gain. The strata use fixed start and goal geometry from the evaluation query, without method outcomes in the bin definitions. Confidence intervals resample the 64 seeds as clusters; query-level paired intervals are used only as a sensitivity analysis. All 512 query keys are paired across the four selectors, and every goal remains inside the same 201-step dataset episode at offset 40.

The stratified results localize the aggregate gain to the 170 object motion queries; both methods solve the 342 static queries. All 27 successes unique to Portfolio ASAR and all 5 successes unique to selection by minimum latent cost occur in the object motion group. Goals with changed contact show the clearest benefit. The larger translation half is a difficult regime: selection by minimum latent cost succeeds on 3/85 and Portfolio ASAR on 0/85, although Portfolio ASAR still improves mean cube position, target position, and yaw errors by .0159, .0036, and .1403. Thus adjacent-set

Table 20: Post hoc Cube stratification using query geometry defined independently of outcomes. Static means object displacement at most 10^{-8}. The 170 moving queries are split at their median XY displacement, 0.1200.

Query stratum	N	Latent minimum	Portfolio	Δ	CI clustered by seed
All queries	512	.723	.766	+.043	[+.021,+.066]
Static object target	342	1.000	1.000	.000	[.000,.000]
Any object motion	170	.165	.294	+.129	[+.063,+.195]
Smaller moving half	85	.294	.588	+.294	[+.176,+.411]
Larger moving half	85	.035	.000	-.035	[-.076,.000]
Stable target contact	400	.897	.905	+.007	[-.005,+.020]
Changed target contact	112	.098	.268	+.170	[+.087,+.248]

Table 21: Targeted reconstruction evaluation. The subset rule uses only start and goal geometry and a planner trace feature from the first replanning step, recorded before outcome inspection. Kernel ASAR improves over least isolated existing sequence selection by .137 in success, with an interval of [+.019,+.238] obtained by resampling seeds, and reduces all three continuous errors.

Method	Success	Cube error	Target pos.	Yaw
Latent minimum	.425	.167	.090	.844
Least isolated	.493	.154	.071	.843
Latent elite mean	.466	.152	.062	.868
Elite mean with minimum anchor	.507	.145	.059	.816
Adjacent prefix mean	.603	.127	.050	.614
Anchored adjacent prefix mean	.603	.136	.053	.737
Kernel ASAR	.630	.135	.047	.762
Portfolio ASAR	**.658**	**.128**	.048	.665

reconstruction often makes progress on long moves but does not provide the missing final placement behavior needed to cross the success threshold.

Continuous error fields. Cube error denotes the evaluator's recorded `final_cube_oracle_distance` quantity. Target position error is `final_block_goal_l2`, a Euclidean position error in meters; yaw error is `final_yaw_goal_abs`, in radians. All recorded values are retained.

Targeted Reconstruction Evaluation. The broader contact motion evaluation did not separate action reconstruction from least isolated existing sequence selection. We therefore recorded a more selective rule before a second evaluation: nonzero yaw change, moderate nonzero object translation, and replan-0 `kernel_max_weight` at most 0.1470682648. This selects cases where the reconstruction weight is distributed across several candidates. On 96 new seeds the rule selects 73 queries across 54 seeds.

I.1 Cube Contact Transition Visualization

The cube contact transition example provides a visualization from the targeted reconstruction setting. The object target requires nonzero translation and yaw change. The initial end-effector contact configuration can also make a single early commitment unreliable. The displayed proposal set uses the same 96 sequences from each of the three fixed proposal scales. Gray curves are cumulative commanded XY action traces. The latent cost minimum is red, the adjacent set is green, and the full sequence reconstructed by Kernel ASAR is blue. Camera images and top views are rendered from simulator execution.

The preselected example set contains six fixed examples. Three are failures under selection by minimum latent cost repaired by Kernel ASAR in the targeted reconstruction evaluation set. One is an agreement success, one is a long motion case where all reconstruction variants fail, and one

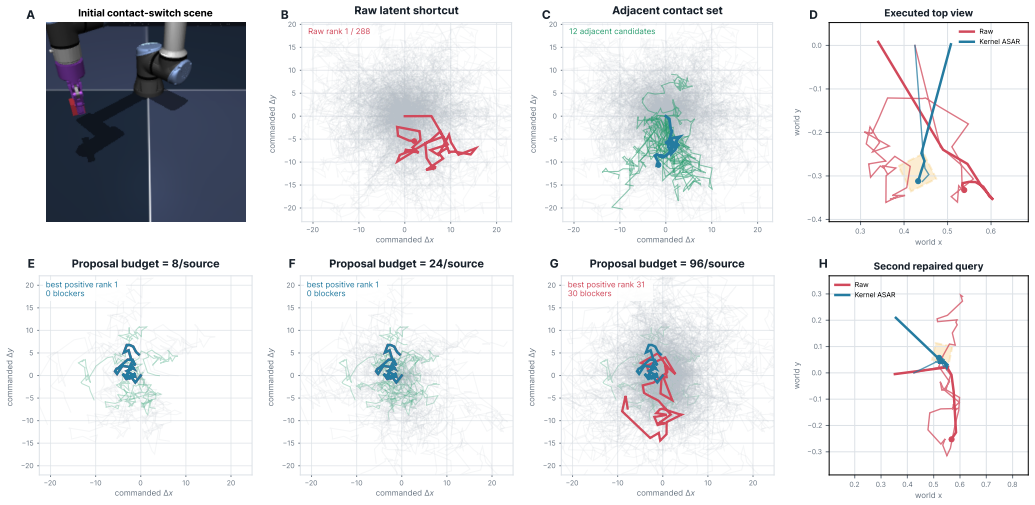


Figure 7: Cube contact transition visualization. (A-D) A failure under selection by minimum latent cost repaired by Kernel ASAR. The minimum cost sequence has an isolated prefix, and the executed top view shows the resulting paths. (E-G) The predetermined proposal budget case from the independent replication. The best feasible position and yaw sequence has latent cost rank 1 at 8 and 24 candidates per source, then falls to rank 31 behind 30 blockers at 96 candidates per source. (H) A second failure under selection by minimum latent cost repaired by Kernel ASAR.

Table 22: MoC Validation of the Mixture of Controllers over 96 new seeds and 768 queries. The comparison is against the first branch, Portfolio ASAR in the same sequential evaluator. MoC rescues 22 Portfolio ASAR failures, loses none, and has a 95th percentile of 4.0 executed branches.

Protocol	Success	Δ	CI	Branches
Portfolio first	.7279	–	–	1.00
MoC	.7565	+.0286	[+.0169,+.0417]	1.77

comes from the independent overgeneration replication. Selection within each category follows deterministic geometry and error improvement rules. These examples are used only for qualitative illustration.

Reconstruction Gains and Mixture of Controllers Verification. Kernel ASAR and Portfolio ASAR succeed on partly different queries. A router based only on start and goal geometry and reconstruction traces available before execution does not reliably choose between them. On all 768 new queries, leave-one-seed-out routing collapses to fixed Kernel ASAR (.7448 versus .7396 for Portfolio ASAR); on the 73-query targeted set, it trails Portfolio ASAR (.6301 versus .6575).

Successful branches are distributed as 559 Portfolio ASAR, 12 Kernel ASAR, 8 anchored adjacent prefix mean, and 2 adjacent prefix mean. The remaining 187 queries have no successful branch among the four controllers. The 22 rescues span 19 distinct seeds. A fixed sequence of four controllers turns this complementarity into a 2.86-point gain when verification with reset is available.

J Replication with an Independently Trained World Model

This experiment changes the world model weights and keeps the Cube environment, query sampler, action normalization, proposal generator, execution budget, and ASAR parameters unchanged. The model is trained with seed 3072 for 100,000 optimization steps on balanced samples from single, double, triple, quadruple, and octuple Cube play data. We use the final available epoch, 32; checkpoint selection is independent of control outcomes. Evaluation uses the separate `ogbench/cube_single_expert` dataset (Park et al., 2025).

Table 23: Replication with independently trained world model weights over 64 seeds and 512 paired queries per budget. Intervals resample evaluation seeds 10,000 times.

Total proposals	Latent minimum	Least isolated	Kernel ASAR
72	.703	.717	**.738**
288	.701	.713	**.727**

Total	Contrast	Δ [95% CI]	W/L
72	Kernel minus latent minimum	+.035 [+.014,+.057]	21/3
72	Kernel minus least isolated	+.021 [+.010,+.035]	11/0
288	Kernel minus latent minimum	+.025 [+.006,+.047]	20/7
288	Kernel minus least isolated	+.014 [-.004,+.031]	13/6

The eligibility pilot contains eight fixed seeds and 64 queries at each budget. Selection by minimum latent cost has 47 successes and 17 failures at 288 total proposals, and Kernel ASAR reconstruction is active in every pool. The predefined evaluation then uses 64 disjoint seeds and eight queries per seed. Both budgets contain the same 512 queries with identical start and goal states. Each source contributes 24 or 96 proposals, giving 72 or 288 proposals in total.

At 72 proposals, reconstruction has positive intervals against both predefined baselines. Its margin over least isolated selection shows that returning one existing sequence is insufficient on this evaluation. Kernel ASAR also improves target position error over least isolated selection by .0034 [.0017,.0052]. Cube position error is unchanged and yaw favors least isolated selection, locating the gain primarily in success and target placement.

At 288 proposals, Kernel ASAR remains better than selection by minimum latent cost in success, cube position error, target position error, and yaw error. Relative to least isolated selection, its success interval crosses zero, while all three continuous error improvements remain positive. Success under selection by minimum latent cost remains stable across budgets, changing from .703 to .701 with a paired difference between the large and small budgets of -.002 [-.020,.016]. Adjacent-set reconstruction therefore transfers across independently trained weights. The different budget trends across checkpoints suggest that the effect of proposal scaling depends on the model's residual prediction errors.

The audit matches all 64 predefined evaluation seeds to the recorded set. Both budgets have 512 identical pool keys and zero mismatches in dataset row, episode, start, goal, seed, or query identifier. Every pool contains one output from selection by minimum latent cost, one from least isolated selection, and one from Kernel ASAR. All runs use the same independently trained checkpoint, and all reported metrics are finite.

K Joint Latent and Action Neighborhood Results and Controls

This section tests whether a joint latent and action neighborhood can preserve a coherent proposal cluster before Portfolio ASAR reconstruction. The first evaluation is used to develop the joint neighborhood rule. The second applies the recorded rule together with random filtering controls matched to the retention rate.

Table 24: Development evaluation of joint neighborhood filtering with 256 paired queries.

Method	Success	Δ latent min.	W/L	Cube error	Target pos.	Yaw
Latent minimum	.6797	.0000	–	.0000	.0000	.0000
Latent elite mean	.7188	.0391	12/2	.0142	.0134	.0716
Anchored adjacent prefix mean	.7227	.0430	12/1	.0123	.0130	.0648
Portfolio ASAR	.7188	.0391	11/1	.0121	.0114	.0723
Latent minimum after graph filtering	.6836	.0039	6/5	-.0045	.0006	-.0629
Anchored mean after graph filtering	.7109	.0312	10/2	.0077	.0103	.0229
Graph-filtered Portfolio ASAR	**.7305**	**.0508**	15/2	.0091	.0096	.0435

Table 25: Confirmatory evaluation of joint neighborhood filtering and matched controls with 256 paired queries.

Method	Success	Δ latent min.	W/L	Cube error	Target pos.	Yaw
Latent minimum	.7344	.0000	–	.0000	.0000	.0000
Anchored adjacent prefix mean	.7344	.0000	5/5	.0083	.0055	.0528
Portfolio ASAR	.7383	.0039	6/5	.0055	.0041	.0433
Graph-Filtered Portfolio ASAR	**.7422**	**.0078**	6/4	.0066	.0032	.0743
Portfolio ASAR with random filtering	.7305	-.0039	5/6	.0042	.0020	.0510
Matched random	.7305	-.0039	5/6	.0054	.0024	.0611

Table 26: Control for joint neighborhood filtering. The random filter matched to the retention rate reproduces the filtering rate but not joint neighborhood centrality or the associated success gain.

Filter	Latent filtered	Size	Latent core	Selected core
Graph component	.6001	40	3.8432	.5797
Random component	.4995	40	3.8942	2.1078
Matched random	.5933	40	3.8842	2.0743

Filtering to the graph core is effective when coupled to adjacent-set reconstruction on the development evaluation. On the second evaluation, latent minimum success is .7344, Portfolio ASAR .7383, Graph-Filtered Portfolio ASAR .7422, and both random controls .7305. The graph increment over Portfolio ASAR is $+.0039$, so graph projection is used as a proposal cluster diagnostic; adjacent-set reconstruction remains the primary method.

L Prefix Density Audit at the Candidate Level

The audit instantiates Proposition 4 using executed candidate pools. It measures pairwise isolation gaps, which are separate from the independent-reference radius statistic and from physical interpolation-ball containment. The ranking blocker is the failed candidate with minimum latent cost inside the 80 lowest cost candidates. The primary feasible candidate is the successful candidate with smallest latent cost in the same pool. The secondary feasible candidate is the successful candidate with the smallest prefix isolation score. The score uses prefix length 1, $k = 3$, robustly standardized kNN distance. The calibration coefficient is selected on discovery seeds as the 0.75 quantile of the required correction and then fixed for held-out seeds and controls.

For the latent minimum pairing, the positive prefix gap rate is .548 and the random assignment test within the pool gives $p = .3605$. Fixed-β coverage is .419 with $p = .2585$. For the pairing with the least isolated feasible candidate, these values rise to .774 ($p = .0032$) and .613 ($p = .0002$). The second pairing explicitly chooses the successful candidate with least isolation. Its higher gap rate describes that selected pairing; the first pairing tests the best-scored feasible candidate.

This distinction matches the method design. ASAR reconstructs a full action sequence anchored to the minimum cost sequence and formed from an adjacent set. It does not rely on scalar pairwise correction alone, and Proposition 4 explains only the reranking part of the mechanism.

M Reacher Transfer Evaluation

The transfer test across tasks uses the official Reacher checkpoint and `dmc/reacher_random`. The Cube Portfolio parameters are used unchanged: 80 lowest cost candidates, prefix length 1, $k = 3$, elite size 7, ratio thresholds 1.0/1.75, and anchor weights on the sequence with minimum cost of 0.25/0.15/0.10. No Reacher threshold or prefix density parameter is tuned. Each of five seeds evaluates 50 unique paired episodes with goal offset 25, budget 50, and CEM configured with 300 samples, 30 iterations, and 30 elites. The supplementary `prefix_roll1` control cyclically permutes prefix feature rows before computing KNN isolation. It preserves the candidate pool and computation while breaking the alignment between each candidate and its prefix features.

Table 27: Prefix density audit on held out data at the candidate level. Each row has 31 pairs, and brackets report 95% intervals.

Pairing and feature	Positive gap	Fixed-β coverage
Failed latent minimum, Prefix-1	.548 [.387,.710]	.419 [.258,.581]
Failed latent minimum, terminal latent	.742 [.581,.871]	.516 [.355,.677]
Failed latent minimum, full sequence	.484 [.323,.645]	.323 [.161,.484]
Failed latent minimum, rolled prefix	.548 [.387,.710]	.194 [.065,.323]
Least isolated feasible, Prefix-1	.774 [.613,.903]	.613 [.452,.774]
Least isolated feasible, terminal latent	.677 [.516,.839]	.419 [.258,.581]
Least isolated feasible, full sequence	.645 [.484,.806]	.484 [.323,.645]
Least isolated feasible, rolled prefix	.581 [.419,.742]	.258 [.129,.419]

Table 28: Reacher transfer with unchanged Cube parameters. The exact paired test gives $p = 1.0$ for Portfolio ASAR against CEM mean.

Method	Success (rate)	Δ	W/L
CEM mean	200/250 (.800)	.000	–
Latent minimum	199/250 (.796)	-.004	28/29
Control with permuted prefixes	198/250 (.792)	-.008	26/28
Portfolio ASAR	199/250 (.796)	-.004	21/22

Method	Episode CI	CI clustered by seed
CEM mean	[.000,.000]	[.000,.000]
Latent minimum	[-.064,.056]	[-.036,.024]
Control with permuted prefixes	[-.064,.048]	[-.064,.048]
Portfolio ASAR	[-.056,.048]	[-.064,.044]

Table 29: Reacher success rates for each seed.

Seed	CEM mean	Latent min.	Permuted prefix	Portfolio
42	.84	.88	.76	.88
43	.76	.70	.82	.78
44	.78	.78	.74	.66
45	.82	.80	.76	.80
46	.80	.82	.88	.86

Portfolio ASAR reaches .796 on Reacher compared with .800 for standard CEM mean, with an interval of [-.064,.044] obtained by resampling seeds. The Reacher candidate pools do not yield the Cube improvement under the same prefix geometry. We next test the same reconstruction rule on Push-T, a task with frequent contact and a nonzero fixed LeWM baseline.

N Push-T Transfer Evaluation

Push-T uses the official fixed LeWM checkpoint and `pusht_expert_train`. The final CEM pool contains 64 candidates after 10 iterations with eight elites. The Cube portfolio is copied without tuning: prefix length 1, $k = 3$, elite size 7, ratio thresholds 1.0/1.75, and anchor weights on the sequence with minimum cost of 0.25/0.15/0.10. Three seeds each evaluate 50 paired episodes at goal offset 50 and budget 50. The protocol and decision rule were recorded before the outcomes, and the reported evaluation uses those settings unchanged.

Table 30: Push-T transfer with unchanged Cube parameters over 150 paired episodes per cost. Positive distance Δ means Portfolio ASAR reduces final task distance.

Cost	Mean	Portfolio ASAR	W/L	Episode CI	Distance Δ
Latent	.273	.280	6/5	[-.040,.053]	+1.95
True state hybrid	.413	.393	6/9	[-.073,.027]	-5.84

Under latent cost, CEM mean reaches .273 and Portfolio ASAR reaches .280 with interval [-.040,.053]. The three paired seed rates are .30/.30, .22/.32, and .30/.22 for CEM mean/Portfolio ASAR. Under true state hybrid cost, Portfolio ASAR loses two points in every seed (.44/.42, .34/.32, and .46/.44) and worsens final distance by 5.84; the distance interval obtained by resampling episodes is $[-9.72, -2.41]$. The latent exact sign test p-value is 1.0 and its success CI clustered by seed is $[-.080, .100]$; the corresponding true state hybrid values are .607 and $[-.020, -.020]$. Portfolio ASAR remains active, using 52 compact, 88 intermediate, and 160 dispersed branches under true state hybrid cost. Contact richness alone does not reproduce the Cube prefix geometry. A transferred prefix metric must also match the task's proposal distribution.

O Alternative Diagnostics and Controls

The comparisons in Table 31 trace the route from latent structure to the final planner intervention. They show which signals are descriptive, which survive closed-loop evaluation, and which component is needed to convert prefix density into task success. The 512-query final proposal pool evaluation gives .711 for latent selection by minimum cost, .721 for latent elite mean, .717 for elite mean with minimum anchor, .732 for adjacent prefix mean, .734 for anchored adjacent prefix mean, .742 for Kernel ASAR, and .738 for Portfolio ASAR. Latent elite averaging has a smaller gain than Kernel ASAR, although the Kernel ASAR success interval against that baseline touches zero.

Table 31: Diagnostic alternatives and their planning outcomes.

Alternative	Outcome and interpretation
Static task rowspace	Variables are decodable, but top-1 success changes only from .323 to .328.
Terminal latent density	Diagnostic structure does not match action prefix reconstruction in closed-loop control.
Index and prefix penalties	Blocker exposure is reproduced without reliable receding-horizon gain.
Latent tangent metrics	Some continuous errors improve, while success trails Kernel ASAR and Portfolio ASAR.
Learned candidate rankers	Gains on held out data vanish under shuffled and composition controls.
Kernel/Portfolio ASAR router	The variants are complementary, but learned switches do not reliably beat the best fixed method.
Graph filtering alone	Selection from the lowest latent costs is nearly flat. Gains require graph projection followed by reconstruction.
Controls for the minimum cost anchor	Alternative anchors retain a partial gain. All effective variants retain adjacent-set reconstruction.

P Implementation and Reproducibility

Paired comparisons reuse the same query identifiers, initial states, evaluation seeds, and proposal keys required by each protocol. Aggregation checks the expected paired keys and finite numeric fields before computing the reported summaries and confidence intervals. Reacher additionally checks identical dataset rows and solver seeds across methods. Push-T preserves paired episode keys and the branch information recorded at each replan.

The confirmatory Cube evaluations used Ubuntu 22.04.5 on x86-64 hardware with two AMD EPYC 7763 processors and two NVIDIA RTX 6000 Ada GPUs. The recorded environment used CUDA 12.8, Python 3.10.20, and PyTorch 2.10.0 with CUDA 12.8 support. The experimental protocols, data splits, controller settings, and statistical procedures used to interpret the reported results are specified in this appendix.