GRAPH-GUIDED LONG-HORIZON MPC WITH LEWORLDMODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Terminal latent distance can poorly reflect progress in long-horizon world-model planning. We add a temporal distance representation, a graph of offline experience, and a budget-conditioned reachability critic to a frozen LeWorldModel (LeWM). Graph search supplies an intermediate target. The critic scores predicted endpoints by the expected time to reach the goal, and the controller switches to latent L2 distance near the goal. We keep the encoder, predictor, and cross-entropy method (CEM) search settings fixed. We bound the number of replanning calls needed to reach the goal under explicit execution, model, search, and critic error conditions. We also bound critic error across horizons and identify when the critic's score equals a truncated expected time to reach the goal. On fixed held-out tasks, success on goals from a different episode increases from 14.4 to 42.4% on Push-T, 29.2 to 65.6% on Reacher, and 16.4 to 27.6% on Cube. Our method shows greatest improvement on long-horizon tasks, while remaining competitive with LeWM on goals 25 steps ahead in Push-T and Reacher. Ablations show that graph guidance and the critic are complementary on Push-T, whereas Cube benefits primarily from the learned temporal representation.

1 INTRODUCTION

Joint-embedding world models enable control by predicting the consequences of candidate actions in a learned latent space (Zhou et al., 2024; Maes et al., 2026b). Their planning objective often compares a predicted terminal latent with an encoded goal image. This avoids learning a task-specific reward, but latent Euclidean distance need not reflect the time or actions required to reach the goal. On our Push-T tasks, the terminal-L2 planner of LeWorldModel (LeWM) succeeds at 91.2% for goals 25 steps ahead, 40.8% at 50 steps, and 13.2% at 100 steps. Cross-episode success is 14.4%, despite replanning every 25 steps.

We use a graph of offline observations to choose intermediate targets. A temporal distance representation (TDR) organizes frozen latents into a graph. Graph search selects a cluster along a shortest route to the goal using a calibrated lookahead distance. The cross-entropy method (CEM) then optimizes a combination of distance to that cluster and a budget-conditioned reachability score. Once the current state is close to the goal in TDR space, the base cost switches to LeWM's latent L2. The critic remains active. We train the TDR and critic while keeping the dynamics model fixed; CEM supplies the actions (Figure 1).

Graph connectivity does not ensure executable routes: proximity edges may be infeasible, and CEM can miss useful actions. We analyze the normalized subgoal and critic cost and prove finite-time progress and local convergence under explicit execution, model, search, and critic error conditions. We also show that reachability values optimized separately for each horizon need not correspond to any single policy's time to reach the goal.

With one shared configuration, the method improves over LeWM for same-episode goals 50 and 100 steps ahead and for cross-episode goals in all three environments. On Cube, however, TDR alone achieves higher cross-episode success than the full method. Our main contributions are:

1. A graph-guided objective for latent model predictive control (MPC), combining dataset subgoals, a remaining-time budget, and a local L2 switch (Section 3).

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.

2. A method-specific progress and convergence analysis, including the effect of CEM approximation and population normalization (Section 4).
3. A controlled comparison on three environments, with component ablations and diagnostics that distinguish long-range guidance from local execution (Section 5).

2 Related work

Costs and hierarchies for latent planning. LeWM and DINO-WM use predictive latent models for visual planning (Maes et al., 2026b; Zhou et al., 2024). TRM learns a temporal pairwise cost on a fixed representation (Li et al., 2026a); RC-aux adds reachability supervision during world-model training (Li et al., 2026b). SAGE uses learned subgoals and action proposals, while Hi-LeWM studies how high-level search interacts with a frozen low-level model (Cheng et al., 2026; Caselli et al., 2026). We choose targets from clusters of dataset observations and keep CEM's action initialization unchanged in the main configuration.

Graph planning and temporal representations. SPTM, SoRB, and SGM combine stored experience with learned local controllers (Savinov et al., 2018; Eysenbach et al., 2019; Emmons et al., 2020). Graph-Assisted Stitching (GAS) uses a temporal graph to guide a learned policy (Baek et al., 2025); SSE addresses subgoal feasibility through replay of execution outcomes (Hwang et al., 2026). We adapt GAS's graph construction to frozen-model MPC. PcLast and ALPS already combine cluster-based planning and CEM; ALPS also learns a behavior prior (Koul et al., 2024; Shehmar et al., 2026). Our contribution combines a frozen pixel world model, a reachability cost that uses the remaining budget, and a local switch. We prove a progress bound for their joint objective. HILP proves goal-reaching results under representation and directional-control errors (Park et al., 2024); quasimetric learning treats the asymmetry of temporal distance explicitly (Wang et al., 2023). We analyze graph attachment, full-plan execution, and cost normalization.

Values and convergence in MPC. POLO and TD-MPC use learned terminal values to complement short model rollouts (Lowrey et al., 2019; Hansen et al., 2022; 2024). Classical MPC stability depends on suitable terminal-region and local-control conditions (Mayne et al., 2000); stochastic shortest-path theory similarly requires conditions ensuring eventual goal arrival (Bertsekas & Tsitsiklis, 1991). Our guarantees likewise rest on explicit control and goal-reaching assumptions.

3 Method

Frozen model and time units. The encoder E maps a 224×224 image to a 192-dimensional latent. The LeWM predictor P uses a three-block history. One action block contains $\Delta = 5$ environment actions; CEM plans $K = 5$ blocks. The controller executes $L = K\Delta = 25$ steps before replanning, unless it reaches the goal first. Every main comparison uses 300 candidates, 30 iterations, and 30 elites. Let $\hat{z}_K$ be the predicted latent after the K planned blocks and $z_g = E(o_g)$ the encoded goal image. LeWM's base cost is $c_{\mathrm{L2}}(\hat{z}_K) = \|\hat{z}_K - z_g\|^2$. We change the scoring function while retaining the encoder, predictor, and search settings. Unless squared explicitly, all norms below are Euclidean distances.

3.1 Temporal representation and graph

Write $d_\psi(z, z') = \|\psi(z) - \psi(z')\|$, with $\psi : \mathbb{R}^{192} \to \mathbb{R}^{32}$. Following HILP and GAS (Park et al., 2024; Baek et al., 2025), two networks learn the negative-distance value $U(z, z_g) = -d_\psi(z, z_g)$ using expectile temporal-difference (TD) learning. For a nonterminal transition (z, z'), the target is $q = -1 + \gamma \min_{i=1,2} U_i^-(z', z_g)$, where $\gamma = 0.99$ and U_i^- are exponential-moving-average (EMA) target networks. Each network U_i minimizes

$$\mathcal{L}_{\mathrm{TDR}} = \mathbb{E}[\ell_{0.99}(q - U_i(z, z_g))], \qquad \ell_\tau(u) = |\tau - \mathbf{1}_{\{u<0\}}|u^2. \tag{1}$$

We set the goal's terminal value to zero. With probability 0.625, we sample a future goal frame at a geometric offset with parameter $1 - \gamma$; otherwise, we sample uniformly from the dataset. The learned distance approximates discounted temporal structure symmetrically. We calibrate its scale using $u_k = \mathrm{median}_t\, d_\psi(z_t, z_{t+k})$ on training trajectories.

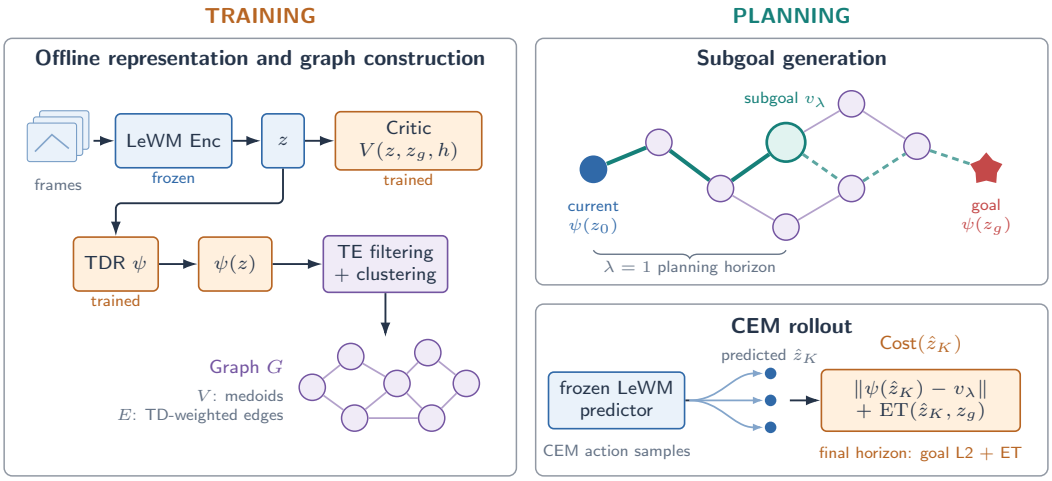


Figure 1: Graph-guided latent MPC. Training (left): from frozen LeWM latents of dataset frames, we train the TDR ψ and the critic V, and build the graph G by clustering in TDR space. Planning (right): a shortest graph path yields the subgoal y_q one planning horizon λ ahead; CEM scores predicted endpoints $\hat{z}_K$ by TDR distance to y_q plus the critic's expected-time (ET) score (Section 3).

Graph construction uses radius $r_{\mathrm{TD}} = u_{12}$ in TDR units and a separate integer temporal lag k_{TE} in environment steps (Table 13). Let F_t be the latent of the first later frame in the same episode at TDR distance at least r_{TD} from z_t, or of the last frame if no such crossing exists. The temporal-efficiency (TE) filter retains frames whose nonzero displacement vectors satisfy

$$\mathrm{TE}(z_t) = \frac{\langle \psi(F_t) - \psi(z_t),\, \psi(z_{t+k_{\mathrm{TE}}}) - \psi(z_t) \rangle}{\|\psi(F_t) - \psi(z_t)\|\, \|\psi(z_{t+k_{\mathrm{TE}}}) - \psi(z_t)\|} \geq 0.9. \tag{2}$$

We clip the lag at the episode end and exclude comparisons with zero displacement. We cluster retained embeddings sequentially, admitting points within $r_{\mathrm{TD}}/2$ of a center and updating centers to member means. Node v stores a TDR center $y_v \in \mathbb{R}^{32}$ and a medoid latent $z_v \in \mathbb{R}^{192}$. We link centers at distance at most r_{TD} with weight $\|y_v - y_w\|$. These links suggest local connections but do not ensure that the dynamics can execute them.

For a goal, a temporary node at $\psi(z_g)$ connects to centers within $r_g = \max(r_{\mathrm{TD}}, 1.2 d_{\min})$, where $d_{\min} = \min_v \|\psi(z_g) - y_v\|$. Dijkstra's algorithm computes $D_g[v]$, the shortest-path length to this node. The resulting graphs have approximately 20.5k, 535, and 16k nodes on Push-T, Reacher, and Cube, respectively. We keep the graph and its distances fixed during each episode.

3.2 Horizon-matched subgoal and local switch

At a planning call with current observation o_t and latent $z_0 = E(o_t)$, restrict attachment to $A(z_0) = \{v : \|\psi(z_0) - y_v\| \leq r_{\mathrm{TD}},\ D_g[v] < \infty\}$ and choose

$$v^\star \in \arg\min_{v \in A(z_0)} \{\|\psi(z_0) - y_v\| + D_g[v]\}. \tag{3}$$

Starting with this attachment segment, follow a shortest path and select the first node whose cumulative path length reaches $\lambda = u_{25}$, or the goal if the path ends earlier. Denote its coordinate by y_q. Selecting the first crossing can overshoot λ by one edge. The theory assumes that an action sequence can reach near this target; median calibration alone does not establish that assumption. The base cost is

$$b(\hat{z}_K) = \begin{cases} \|\hat{z}_K - z_g\|^2, & d_\psi(z_0, z_g) \leq \lambda, \\ \|\psi(\hat{z}_K) - y_q\|, & d_\psi(z_0, z_g) > \lambda. \end{cases} \tag{4}$$

The controller recomputes the switching decision at each call. The graph branch requires $A(z_0) \neq \varnothing$; disconnected or unattached states have no finite route to follow. The controller uses the nearest node as a fallback in this case, but the graph-progress theorem does not cover that fallback.

3.3 REACHABILITY SCORE AND THE REMAINING BUDGET

We query the five-member critic ensemble $V(z, z_g, h) \in [0, 1]$ at $h = 0, 5, \ldots, 50$ environment steps. We train it with binary cross-entropy on hindsight first-hit labels $\mathbf{1}_{\{k^\star \le h\}}$, cross-episode negatives, and predicted latents under logged actions. For a state z_t, $k^\star = \min\{k \ge 0 : d_\psi(z_{t+k}, z_g) < u_2\}$ is the first logged step at which the trajectory is within TDR distance u_2 of the goal. Cross-episode pairs receive label 0 only when $d_\psi(z, z_g) \ge 1.25\, u_{\max(h,5)}$, that is, when the TDR places the goal beyond budget h. No label reads the simulator state, so training and inference are both pixel-only; on Push-T, the TDR-derived $k^\star$ agrees with the success predicate on 97.7% of draws. For $h \ge \Delta$ at unsuccessful states, we also form a bootstrapped target from $M = 8$ action blocks (the logged block, perturbations of it, and random blocks). With $\hat{z}'_m$ the predicted next latent under block m, and $\bar{V}, \bar{\sigma}$ the mean and standard deviation of the EMA target ensemble,

$$q_h = \max_{m \le M} \mathrm{clip}_{[0,1]}\{\bar{V}(\hat{z}'_m, z_g, h - \Delta) - \kappa \bar{\sigma}(\hat{z}'_m, z_g, h - \Delta)\}, \quad \kappa = 1. \tag{5}$$

A hinge loss penalizes $[V(z, z_g, h) - V(z, z_g, h + \Delta)]_+$, enforcing monotonicity in h. These labels and the ensemble penalty are heuristics: a TDR ball only approximates the goal condition, and ensemble spread is not a calibrated confidence interval. We therefore treat the critic as an approximate reachability score and account for its error in Section 4.

Let B be the budget and t the executed steps. For a full plan with $B - t \ge L$, set $h_{\mathrm{rem}} = \min(50, B - t - L)$ and $n = \lfloor h_{\mathrm{rem}}/\Delta \rfloor$. We sum over all grid points, including both endpoints:

$$\mathrm{ET}(\hat{z}_K, z_g) = \Delta \sum_{j=0}^{n} [1 - V(\hat{z}_K, z_g, j\Delta)], \qquad C_n = (n + 1)\Delta, \tag{6}$$

which we call the expected-time (ET) score. Each term $1 - V$ is a predicted *survival* probability, that of not yet having reached the goal by $j\Delta$ further steps; the score is capped at C_n, one grid interval beyond the largest queried deadline. When no time remains, the score still penalizes predicted failure at the endpoint. The score equals a truncated expected *grid* hitting time only when all queried probabilities describe the same continuation policy. Otherwise, we use it as a sum of reachability estimates. Updating the budget prevents queries beyond the available time; it does not guarantee success by the deadline.

At each CEM iteration, we normalize each cost over the candidate population using $\widetilde{f} = (f - \mathrm{median}\, f)/s_f$, where $s_f = \max(\mathrm{IQR}\, f, \nu_f)$ and IQR is the interquartile range. The fixed positive floor ν_f prevents division by zero. Writing $S = \mathrm{ET}(\hat{z}_K, z_g)$ for the critic score, the objective is

$$c = \widetilde{b} + \beta \widetilde{S}, \qquad \beta = 1. \tag{7}$$

Although β is fixed, the effective relative weight in unnormalized units is $\beta s_b / s_S$. This ratio is central to the progress bound. The TDR and critic train for 50k and 60k updates; Appendices A and H give the remaining settings and planning pseudocode.

4 THEORY: FROM THE PLANNING COST TO GOAL PROGRESS

We analyze the normalized graph and critic cost with the trained networks and graph held fixed. Our main result bounds graph progress and local convergence without assuming a globally accurate temporal metric. Appendix I connects each assumption to the algorithm and provides the proofs. Fix a goal g. Define the remaining graph distance Φ by the attachment objective,

$$\Phi(z) = \min_{v \in A(z)} \{\|\psi(z) - y_v\| + D_g[v]\}, \qquad \mathcal{R}_\lambda = \{z : d_\psi(z, z_g) \le \lambda\}. \tag{8}$$

For deterministic execution, we analyze CEM's returned sequence using the predictor's input history. The controller ends the episode as soon as the agent satisfies the environment's goal condition. Appendix I.5 covers stochastic execution.

Theorem 1 (Normalized-cost graph progress)**.** *Suppose $z \notin \mathcal{R}_\lambda$ has a finite attachment, and let y_q be the first λ-crossing target on its selected path. Write $z^+(u), \hat{z}^+(u)$ for the true and predicted endpoints of an L-step sequence. Suppose a feasible comparison sequence u° satisfies $\|\psi(z^+(u^\circ)) - y_q\| \le \eta$. For both u° and the executed sequence $\hat{u}$, assume $\|\psi(\hat{z}^+(u)) - \psi(z^+(u))\| \le e_\psi$. Let*

$\hat{c}(u)$ be the objective Equation (7) at $\hat{z}^+(u)$, normalized with the final population's medians and positive scales s_b, s_S, and $\widehat{S}(u)$ the unnormalized critic score there. Suppose $\hat{c}(\hat{u}) \le \hat{c}(u^\circ) + \epsilon_{\mathrm{opt}}$ and $[\widehat{S}(u^\circ) - \widehat{S}(\hat{u})]_+ \le \omega$, and define

$$\rho = \eta + 2e_\psi + s_b\epsilon_{\mathrm{opt}} + \beta\frac{s_b}{s_S}\omega. \tag{9}$$

If $\rho \le r_{\mathrm{TD}}$ and $\rho < \lambda$, the controller either reaches the goal during the call, enters $\mathcal{R}_\lambda$, or reduces the remaining graph distance:

$$\Phi(z^+(\hat{u})) \le \Phi(z) - (\lambda - \rho). \tag{10}$$

If these bounds hold at every graph call before stopping, success or entry into $\mathcal{R}_\lambda$ occurs within $N_G = \lceil \Phi(z_0)/(\lambda - \rho) \rceil$ calls.

An endpoint within r_{TD} of a regular target node can attach there for the next call, reducing the remaining graph distance by at least $\lambda - \rho$. If the target is the goal node, the endpoint enters the local region instead. The bound includes the normalization ratio in Equation (9); $\omega = C_n$ always bounds the critic difference but may be too large to prove progress. Median calibration alone does not ensure that the comparison sequence exists. Proposition 19 also shows when the critic rejects poor plans that subgoal distance alone would favor.

Theorem 2 (Local convergence after the switch)**.** *Assume the switched controller stays in $\mathcal{R}_\lambda$ until success and enters with $e(z) = \|z - z_g\| \le R$. Assume $e(z) \le r_{\mathrm{goal}}$ implies the environment's goal condition. At each local call, suppose a feasible comparison sequence reduces true latent error to at most $ae(z)$, where $0 < a < 1$. Suppose endpoint prediction error is at most e_z for the comparison and executed sequences, and the normalized solver and critic bounds of Theorem 1 hold with local constants. Put*

$$d_\ell = 2e_z + \sqrt{s_b\epsilon_{\mathrm{opt}} + \beta(s_b/s_S)\omega_\ell}, \qquad e_\infty = \frac{d_\ell}{1-a}. \tag{11}$$

Let z_m be the latent after m local calls. Until success, $e(z_m) \le a^m R + (1-a^m)e_\infty$. If $e_\infty < r_{\mathrm{goal}}$, success follows within

$$N_\ell = \max\left\{0, \left\lceil \frac{\log(R/(r_{\mathrm{goal}} - e_\infty))}{\log(1/a)} \right\rceil \right\}$$

local calls, with $N_\ell = 0$ if $R = 0$. Together with Theorem 1, $L(N_G + N_\ell)$ is a sufficient execution budget, provided the bounds hold for every remaining-budget value encountered.

These assumptions require the local controller to stay in the switching region and reduce latent error enough to satisfy the task's goal condition. They play the same role as local control and terminal-region conditions in MPC (Mayne et al., 2000). The reported success rates alone do not establish them.

Theorem 3 (Finite-horizon critic error propagation)**.** *Use a Markov state sufficient to specify block-transition probabilities and the goal condition. After success, keep the process in a successful state. Let V_j^* be the optimal probability of reaching the goal by j blocks: $V_0^* = \mathbf{1}_g$, the indicator of the goal condition, and $V_j^* = \mathcal{B}V_{j-1}^*$, where $\mathcal{B}$ is the undiscounted reachability Bellman operator. For estimates $\hat{V}_j$, suppose $\|\hat{V}_0 - V_0^*\|_\infty \le \varepsilon_0$ and $\|\hat{V}_j - \mathcal{B}\hat{V}_{j-1}\|_\infty \le \xi_j$. Then*

$$\|\hat{V}_j - V_j^*\|_\infty \le \varepsilon_0 + \sum_{i=1}^{j}\xi_i,$$

$$\left|\Delta\sum_{j=0}^{n}(1-\hat{V}_j) - \Delta\sum_{j=0}^{n}(1-V_j^*)\right| \le \Delta\left[(n+1)\varepsilon_0 + \sum_{i=1}^{n}(n-i+1)\xi_i\right]. \tag{12}$$

Exact synchronous backups with the boundary held fixed recover horizons through n after n sweeps, from any bounded initialization of the other horizons.

We apply standard reachability dynamic programming (Abate et al., 2008) to this critic. Its residuals include model, action-search, label, and fitting errors; the training loss alone does not bound them. For a common continuation policy π, the sum of survival probabilities is exactly $\mathbb{E}_\pi \min(T^\pi, C_n)$ for grid hitting time T^π. For horizon-wise optimal values it is only a lower bound on $\inf_\pi \mathbb{E}_\pi \min(T^\pi, C_n)$ (Proposition 15). Thus ET need not describe any single policy or guarantee success by the deadline. Appendix I.9 proves the separate tabular expectile convergence result and explains why it does not imply neural TDR convergence.

Why improvement over terminal L2 is possible. Proposition 25 constructs a problem with an exact model and latents on a circle. A K-block L2 planner stays at the start forever, while the graph objective with an exact critic reaches the goal in two calls, including $K = 5$. Reaching the goal requires a temporary increase in L2 distance, which the short-horizon L2 planner avoids. This example explains one advantage; incorrect edges, poor action coverage, or a conflicting critic can reverse it.

5 EXPERIMENTS

5.1 SETUP

Environments and protocols. We use the public LeWM datasets and checkpoints: 18,685 Push-T expert episodes (2.34M frames), and 10,000 episodes of 201 frames each for Reacher and OGBench Cube (Maes et al., 2026b; Park et al., 2025). Push-T success requires agent and block position errors below 20 px and block angle error below $\pi/9$. Reacher uses raw joint-angle differences below 0.05 rad for both joints, without angle wrapping. Cube requires cube-position error below 0.04 m, without an orientation or end-effector condition. The Reacher convention distinguishes unwrapped joint coordinates and therefore need not be identifiable from a single image.

Same-episode goals lie at offset $H \in \{25, 50, 100\}$ environment steps after the start in the same trajectory, with budget $B = 2H$. Cross-episode tasks draw frames from different episodes, reject pairs already successful at the start, and impose no additional distance or reachability filter. Their budgets are 250 steps on Push-T and 200 on Reacher and Cube. Each protocol uses the same 50 evaluation tasks across methods and five seeds. These tasks are held out: their episodes are excluded from TDR, graph, and critic training and calibration. For our planner, seeds vary TDR, graph, critic, and CEM randomness; L2 seeds vary CEM alone. Standard deviations measure variation across seeds on these fixed tasks. They do not measure uncertainty across new tasks or establish generalization to unseen goals.

Baselines. L2 uses LeWM's terminal cost. TDR uses $d_\psi(\hat{z}_K, z_g)$ directly. The graph cost-to-go (CTG) baseline uses a terminal attachment cost plus D_g without committing to a subgoal. Random samples environment actions uniformly. We train GCIQL (Kostrikov et al., 2022; Park et al., 2025) on frozen LeWM latents with the configuration used for LeWM's baselines (Maes et al., 2026a), ten epochs for each of its value and policy phases, and one training seed per environment. All planning comparisons use the same main CEM configuration. Panel (B) of Table 1 gives published results for context. Those evaluations use different tasks from our matched comparison.

5.2 MAIN RESULTS AND COMPONENT EFFECTS

Table 1 shows gains over LeWM at offsets 50 and 100 and on cross-episode goals in every environment. Push-T gains are 29.2, 33.2, and 28.0 percentage points; Reacher gains are 7.2, 14.8, and 36.4; Cube gains are 15.2, 14.8, and 11.2. At offset 25, the method loses 2.8 points on Push-T and 1.2 on Reacher, and gains 6.0 on Cube. GCIQL is below the method at every setting in every environment (cross-episode 2.0%, 0.0%, and 12.0% on Push-T, Reacher, and Cube).

Push-T ablations. Table 2 isolates the components on the same tasks. TDR alone improves offsets 50 and 100 but leaves cross-episode success near L2. Adding graph subgoals and the local switch reaches 62.4/30.8/17.2% at offsets 50/100/cross-episode. Adding the ET critic to L2 without a graph reaches 58.4/38.4/31.6%; combining it with the graph and switch reaches 70.0/46.4/42.4%. Thus the graph and critic are complementary on this environment. Switching to L2 locally recovers much of the short-range loss and adds 7.2 points cross-episode when the critic is present.

The variant (f), which replaces ET by $-\log V(\hat{z}_K, z_g, h_{\mathrm{rem}})$, has the largest offset-100 mean (47.2%), slightly above ET (46.4%), but loses 10.8 points to ET at offset 25 and 8.0 cross-episode. Integrating survival over several horizons can distinguish candidates that have similar probability at one deadline (Proposition 14). This explanation requires informative horizon predictions; a saturated or miscalibrated critic need not help. The weight sweep in Table 6 also shows that $\beta = 1$ is a shared choice rather than an optimal value for every protocol.

Table 1: Success rate (%). **(A)** Our evaluation on the fixed, held-out tasks, mean ± sd over five seeds. Within each environment, **bold** is the best mean and underlined means having a ±sd interval that overlaps the best's. †One training seed with a deterministic policy, so no sd is reported; see Section 5.1. **(B)** Numbers reported by Maes et al. (2026b) for LeWM, PLDM (Sobal et al., 2025), DINO-WM (Zhou et al., 2024), and the offline goal-conditioned reinforcement learning baselines GCBC, GCIQL, and GCIVL (Park et al., 2025). These were not run on our task set, which rejects pairs already solved at the start, so they are not directly comparable to (A). "–": not reported.

(A) Our results

Env	Method	same-ep 25	same-ep 50	same-ep 100	cross-ep
Push-T	Random	4.0 ± 1.3	0.4 ± 0.8	0.0 ± 0.0	0.4 ± 0.8
	GCIQL†	42.0	26.0	22.0	2.0
	L2 (LeWM)	**91.2** ± 1.0	40.8 ± 3.7	13.2 ± 3.2	14.4 ± 3.9
	Ours	88.4 ± 2.3	**70.0** ± 2.2	**46.4** ± 3.2	**42.4** ± 3.2
Reacher	Random	8.0 ± 3.8	8.4 ± 0.8	9.2 ± 3.2	2.8 ± 1.0
	GCIQL†	0.0	0.0	2.0	0.0
	L2 (LeWM)	**77.2** ± 4.8	90.8 ± 1.0	82.8 ± 1.0	29.2 ± 2.4
	Ours	76.0 ± 4.6	**98.0** ± 1.3	**97.6** ± 0.8	**65.6** ± 2.0
Cube	Random	21.2 ± 3.0	12.8 ± 2.0	4.8 ± 4.1	3.6 ± 2.3
	GCIQL†	32.0	28.0	28.0	12.0
	L2 (LeWM)	56.0 ± 3.8	34.4 ± 3.2	35.6 ± 1.5	16.4 ± 1.5
	Ours	**62.0** ± 4.7	**49.6** ± 4.6	**50.4** ± 3.2	**27.6** ± 6.1

(B) Reported by Maes et al. (2026b) (same-ep 25 only)

Env	LeWM	PLDM	DINO-WM	GCBC	GCIQL	GCIVL	Ours
Push-T	96	78	74	75	20	33	88.4
Reacher	86	78	79	–	–	–	76.0
Cube	74	65	86	84	64	56	62.0

Table 2: Push-T ablation (success %, same tasks and seeds as Table 1). Rows (a)–(c) build the graph subgoal without a critic; (d) adds the critic without a graph; (e)–(g) add a critic to the graph subgoal; (h) scores the subgoal with LeWM's L2 instead of TDR distance. Full ladder in Table 6. **Bold**: best mean per column; underlined: ±sd interval overlaps the best's.

	configuration	same-ep 25	same-ep 50	same-ep 100	cross-ep
	L2 (LeWM)	**91.2** ± 1.0	40.8 ± 3.7	13.2 ± 3.2	14.4 ± 3.9
(a)	TDR terminal cost (no graph)	83.2 ± 4.1	48.4 ± 1.5	22.4 ± 5.3	14.0 ± 1.3
(b)	graph subgoal	80.8 ± 2.7	51.6 ± 2.3	22.4 ± 5.0	15.2 ± 3.0
(c)	(b) + local switch	88.0 ± 2.5	62.4 ± 2.7	30.8 ± 6.5	17.2 ± 2.7
(d)	L2 + ET critic (no graph)	89.2 ± 3.0	58.4 ± 5.1	38.4 ± 7.1	31.6 ± 2.3
(e)	(b) + ET critic (no switch)	86.0 ± 3.4	64.8 ± 3.0	43.6 ± 5.0	35.2 ± 2.0
(f)	(c) + $-\log V$ critic	77.6 ± 1.5	69.2 ± 2.4	**47.2** ± 5.9	34.4 ± 7.3
(g)	(c) + ET critic (**ours**)	88.4 ± 2.3	**70.0** ± 2.2	46.4 ± 3.2	**42.4** ± 3.2
(h)	(g), subgoal scored by L2	86.0 ± 2.2	63.6 ± 4.8	45.2 ± 3.5	36.4 ± 2.9

Environment dependence. The Cube ablation in Table 7 shows a different pattern: TDR alone reaches 36.8% cross-episode, exceeding the full method's 27.6%. Removing the local switch gives the highest offset-100 mean (56.8%). Therefore the full method's improvement over LeWM on Cube does not establish that graph routing or the critic improves on TDR. Theorem 1 is consistent with this outcome: it requires executable targets and bounds how strongly the critic may oppose target tracking, and neither condition is guaranteed on Cube.

Table 3: Push-T success (%), mean over five seeds. **(a)** Success under loosened position/angle tolerances; the default is 20 px/20°. **(b)** Cumulative success at the end of each 50-step budget window; in brackets, the hazard rate, i.e., the share of tasks unsolved at the start of the window that are solved within it. Same-ep 100 has a 200-step budget, hence "–" for entries at higher budgets.

(a) Success threshold

threshold	Ours	L2
same-ep 100		
20 px/20°	46.4	13.2
25 px/25°	52.8	14.8
30 px/30°	58.0	16.0
cross-ep		
20 px/20°	42.4	14.4
25 px/25°	51.2	19.2
30 px/30°	57.6	22.8

(b) Success by budget window

window	same-ep 100		cross-ep	
(steps)	Ours	L2	Ours	L2
0–50	8.8 (8.8)	8.8 (8.8)	14.8 (14.8)	6.0 (6.0)
50–100	23.2 (15.8)	12.0 (3.5)	30.4 (18.3)	12.0 (6.4)
100–150	36.4 (17.2)	12.4 (0.5)	36.4 (8.6)	13.2 (1.4)
150–200	46.4 (15.7)	13.2 (0.9)	38.8 (3.8)	14.0 (0.9)
200–250	–	–	42.4 (5.9)	14.4 (0.5)

5.3 Diagnostics and computational cost

The Push-T diagnostics distinguish model error from objective ranking. Under logged actions, median predictor error is 2.41 at 25 steps and 7.80 at 100 steps; true start–goal distances are 17.06 and 19.60, respectively (Table 8 and Figure 3). For L2, 25 steps is at or near the best open-loop plan length, and 100-step plans lower success (Table 9). Because the true start–goal distance already reaches 19.45 at 50 steps, close to the random-pair average of 19.6, terminal L2 carries little information about temporal progress beyond one planning horizon.

To test ranking directly, we compare each cost with a simulator-based recovery time (the number of steps an oracle controller needs to reach the goal from a candidate's true endpoint; Appendix F). Over our method's first-iteration CEM elites on 25 tasks, mean Spearman correlations are 0.327/0.203/0.191 at offsets 25/50/100, versus 0.396/0.086/-0.064 for L2 (Table 10). L2 ranks candidates better at 25 steps and near chance beyond, which is consistent with switching to L2 near the goal. L2 also rewards endpoints near the straight latent line to the goal. Encoded frames lie on a thin shell of radius about 13.9 (Appendix E.3), whereas straight-line midpoints between start and goal latents 100 steps apart have mean radius 10.03 (Figure 2). The straight latent path therefore passes through a region that observed trajectories do not visit.

Table 3 shows that our cumulative Push-T success continues to rise over later replans. On cross-episode tasks, success grows from 30.4% at step 100 to 42.4% at step 250, versus 12.0 to 14.4% for L2: later replans keep solving tasks under our method, whereas L2 rarely solves a task after the first 100 steps. Loosening the success tolerances raises both methods' scores and widens our advantage, so more of our failures are near misses. Demonstration initialization is a useful additional option on Cube: it raises cross-episode success from 27.6 to 48.0%, but helps neither Push-T nor Reacher (Table 4 and Appendix C). This variant initializes CEM's mean with a 25-step logged action sequence whose start is close to the current state and whose endpoint is close to the graph target in TDR space. It retains the cost, candidate count, and execution cadence. The differences across environments show that planning depends on both the sampled actions and the cost used to rank them.

On one GTX 1080 Ti, preparation takes 190.8 GPU-minutes including shared encoding; planning takes 1.80 seconds per decision versus 1.14 for L2, an increase of 58% (Table 5). We run Dijkstra's algorithm once per goal. The deployed TDR and critic add 0.64M and 4.13M parameters (Table 11).

6 Limitations

The critic's labels come from TDR distances, so their accuracy depends on the TDR: a TDR ball can disagree with the success predicate, and heuristic negatives can mislabel reachable pairs. Label errors, off-distribution model rollouts, finite action proposals, and representation aliasing can violate the theoretical error conditions. The graph links nearby centers and connects the goal using an expanded radius, but these distances do not ensure feasible transitions. The convergence result

Table 4: Demonstration retrieval added to our method, with no other change (success %, same 50 held-out tasks and five seeds as Table 1, mean ± sd). **Bold**: better mean of the two rows. Details in Appendix C.

Env	Method	same-ep 25	same-ep 50	same-ep 100	cross-ep
Push-T	Ours	**88.4** ± 2.3	**70.0** ± 2.2	**46.4** ± 3.2	42.4 ± 3.2
	Ours + retrieval	86.8 ± 2.0	66.8 ± 5.2	44.8 ± 4.7	42.4 ± 6.2
Reacher	Ours	**76.0** ± 4.6	**98.0** ± 1.3	**97.6** ± 0.8	**65.6** ± 2.0
	Ours + retrieval	62.0 ± 6.4	88.4 ± 2.7	94.0 ± 1.3	63.6 ± 2.3
Cube	Ours	62.0 ± 4.7	49.6 ± 4.6	50.4 ± 3.2	27.6 ± 6.1
	Ours + retrieval	**86.4** ± 4.1	**59.2** ± 2.4	**78.8** ± 3.0	**48.0** ± 2.8

Table 5: Push-T compute cost on one GTX 1080 Ti. Preparation is a one-off ∼3.2 h, mostly critic training; planning with the graph and critic takes 1.6× L2's time per decision.

One-off preparation (GPU-min, per seed)	
Encode (shared across seeds)	49.6
TDR training (50k steps)	16.6
Graph construction	1.6
Critic training (60k steps)	123.0
Total (incl. shared encode)	190.8
Per-decision planning time (s)	
L2	1.14
Ours (graph subgoal + critic)	1.80
Ratio vs. L2	1.58×

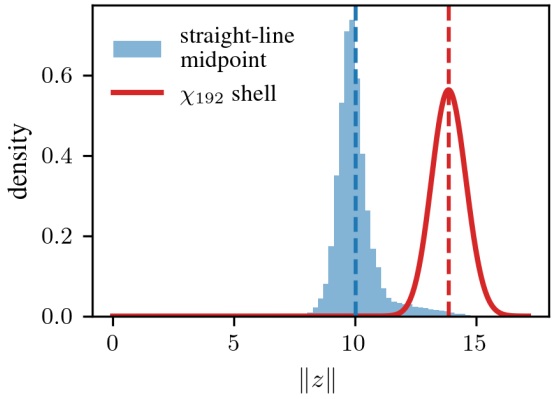


Figure 2: Straight-line midpoints leave the data distribution. Push-T, same-episode pairs $(z_{\text{start}}, z_{\text{goal}})$, 100 steps apart: radius of midpoints $\|\frac{1}{2}(z_{\text{start}} + z_{\text{goal}})\|$ (blue) vs. the χ_{192} shell that encoded frames lie in (red).

assumes local control and that sufficiently small latent error implies task success. It does not certify the released checkpoints or prove global convergence of neural training or CEM.

Experiments cover one world-model family and fixed task sets. They do not include matched runs of SAGE, Hi-LeWM, or ALPS. SAGE reports strong results at offset 150 (Cheng et al., 2026), a setting outside our main comparison. We claim neither overall state of the art nor a general advantage over learned subgoal generators. The Cube ablations and short-range losses show where the combined objective fails to help.

7 Conclusion

Terminal latent L2 ranks CEM candidates well within one planning horizon but poorly beyond it. We replaced only the planning cost of a frozen LeWM: a graph subgoal one horizon along a shortest path through offline data, a budget-aware expected-hitting-time critic, and a switch to L2 near the goal. With the encoder, predictor, and CEM settings unchanged, this improves long-horizon and cross-episode success in three environments, and our analysis states the execution, model, search, and critic conditions under which such progress is guaranteed. The components are complementary on Push-T, whereas on Cube most of the gain comes from the temporal representation.

AI use statement

In this work, we used generative AI tools (an LLM-based coding assistant) to write and refactor implementation and analysis code, to help with literature search, mathematical derivations, proof-

checking, and manuscript revision. We have not used generative AI tools to generate synthetic data, to develop the conceptual framework, or to formulate mathematical claims. We have reviewed all AI-assisted work: LLM-written code was read and tested by the authors, every reported number was produced by runs the authors launched and checked, and citations found with AI assistance were verified against the original papers. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

This work uses simulated benchmarks with public datasets and checkpoints and involves no human subjects or sensitive data.

Reproducibility statement

We will release anonymised code as supplementary material. It covers dataset encoding, TDR training, graph construction, critic training, and CEM evaluation, together with the driver scripts used to produce every table. All experiments use the public LeWM checkpoints and datasets (Maes et al., 2026b). Episodes that contain evaluation tasks are excluded from all learned-asset training and calibration. Section 5.1 describes the protocols and seeds, Appendix A the planning procedure, Appendix H all hyperparameters (including the unchanged CEM configuration), and Table 5 compute costs. Live frames must be rendered with the MuJoCo version that produced the dataset (Appendix D); we pin MuJoCo 3.5.0 and dm_control 1.0.37 for Reacher.

References

Alessandro Abate, Maria Prandini, John Lygeros, and Shankar Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008. doi: 10.1016/j.automatica.2008.03.027. URL https://www.cs.ox.ac.uk/people/alessandro.abate/publications/APLS08.pdf.

Seungho Baek, Taegeon Park, Jongchan Park, Seungjun Oh, and Yusung Kim. Graph-assisted stitching for offline hierarchical reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2025. URL https://proceedings.mlr.press/v267/baek25a.html. arXiv:2506.07744.

Dimitri P. Bertsekas and John N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991. doi: 10.1287/moor.16.3.580. URL https://www.mit.edu/~jnt/Papers/J034-91-berts-SSP.pdf.

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

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.

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

Scott Emmons, Ajay Jain, Michael Laskin, Thanard Kurutach, Pieter Abbeel, and Deepak Pathak. Sparse graphical memory for robust planning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2003.06417.

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

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://sites.stat.washington.edu/people/raftery/Research/PDF/Gneiting2007jasa.pdf.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning (ICML)*, 2022. URL https://arxiv.org/abs/2203.04955.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.16828.

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. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500830.

Jaebak Hwang, Sanghyeon Lee, Jeongmo Kim, and Seungyul Han. Strict subgoal execution: Reliable long-horizon planning in hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2506.21039.

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://proceedings.neurips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2110.06169.

Anurag Koul, Shivakanth Sujit, Shaoru Chen, Ben Evans, Lili Wu, Byron Xu, Rajan Chari, Riashat Islam, Raihan Seraj, Yonathan Efroni, Lekan P Molu, Miroslav Dudík, John Langford, and Alex Lamb. PcLast: Discovering plannable continuous latent states. In *International Conference on Machine Learning*, pp. 25475–25493, 2024. URL https://proceedings.mlr.press/v235/koul24a.html.

Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

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

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.

Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan online, learn offline: Efficient learning and exploration via model-based control. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1811.01848.

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.

David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. doi: 10.1016/S0005-1098(99)00214-9. URL https://www.sciencedirect.com/science/article/pii/S0005109899002149.

Whitney K. Newey and James L. Powell. Asymmetric least squares estimation and testing. *Econometrica*, 55(4):819–847, 1987. doi: 10.2307/1911031. URL https://www.jstor.org/stable/1911031.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with Hilbert representations. In *International Conference on Machine Learning (ICML)*, 2024. URL https://proceedings.mlr.press/v235/park24g.html.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.20092.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://arxiv.org/abs/1803.00653.

Dikshant Shehmar, Matthew Schlegel, Matthew E. Taylor, and Marlos C. Machado. Laplacian representations for decision-time planning. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.05031.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2502.14819.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2304.01203.

J. G. Wendel. Note on the gamma function. *The American Mathematical Monthly*, 55(9):563–564, 1948.

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

A PLANNING ALGORITHM

Algorithm 1 One planning call of graph-assisted latent MPC

Require: frozen E, P; CEM; TDR ψ; graph (V_G, c_v, D_g); critic V; budget B; step t

$z_0 \leftarrow E(o_t)$; $h_{\mathrm{rem}} \leftarrow \min(h_{\max}, B - t - 5K)$
if $\|\psi(z_0) - \psi(z_g)\| \le \lambda$ **then**
  target $\leftarrow$ goal; base cost $\leftarrow c_{\mathrm{L2}}$
else
  $v^\star \leftarrow \arg\min_{v:\|\psi(z_0)-c_v\|\le r_{\mathrm{TD}}} D_g[v] + \|\psi(z_0) - c_v\|$
  walk the shortest path from $v^\star$ until cumulative TDR distance $\ge \lambda$; target $\leftarrow v_\lambda$; base cost $\leftarrow c_{\mathrm{sub}}$ (goal and c_{L2} if the walk reaches the goal first)
end if
for each CEM iteration **do**
  sample 300 action-block sequences; $\hat{z}_{1..K} \leftarrow P(\cdot)$
  $c \leftarrow \widetilde{\mathrm{base}}(\hat{z}_K) + \beta\, \widetilde{\mathrm{ET}}(\hat{z}_K, z_g)$ with ET from Equation (6)
  refit the CEM Gaussian on the 30 lowest-c candidates
end for
execute the mean sequence for $5K$ steps; $t \leftarrow t + 5K$

B ABLATION LADDER

B.1 PUSH-T

Table 6: Full Push-T ablation ladder (success %, same tasks and seeds as Table 1, mean $\pm$ sd over five seeds). Letters (a)–(h) match Table 2. The lower blocks vary the critic weight β and the graph threshold r_{TD} (in TDR units) of the full method (g), with every other setting unchanged. The graph cost-to-go row uses the graph distance as a terminal value instead of a subgoal; its cost is $\min(\text{direct TDR distance}, \text{graph distance})$, so it departs from (a) only when the graph route is shorter. At same-ep 25 and cross-ep the two rows are separate runs that solve the same tasks on every seed.

	configuration	same-ep 25	same-ep 50	same-ep 100	cross-ep
	L2 (LeWM)	91.2 ± 1.0	40.8 ± 3.7	13.2 ± 3.2	14.4 ± 3.9
(a)	TDR terminal cost (no graph)	83.2 ± 4.1	48.4 ± 1.5	22.4 ± 5.3	14.0 ± 1.3
	graph cost-to-go (no subgoal)	83.2 ± 4.1	47.2 ± 1.6	21.6 ± 4.6	14.0 ± 1.3
(b)	graph subgoal	80.8 ± 2.7	51.6 ± 2.3	22.4 ± 5.0	15.2 ± 3.0
(c)	(b) + local switch	88.0 ± 2.5	62.4 ± 2.7	30.8 ± 6.5	17.2 ± 2.7
(d)	L2 + ET critic (no graph)	89.2 ± 3.0	58.4 ± 5.1	38.4 ± 7.1	31.6 ± 2.3
(e)	(b) + ET critic (no switch)	86.0 ± 3.4	64.8 ± 3.0	43.6 ± 5.0	35.2 ± 2.0
(f)	(c) + $-\log V$ critic	77.6 ± 1.5	69.2 ± 2.4	47.2 ± 5.9	34.4 ± 7.3
(g)	(c) + ET critic (**Ours**)	88.4 ± 2.3	70.0 ± 2.2	46.4 ± 3.2	42.4 ± 3.2
(h)	(g), subgoal scored by L2	86.0 ± 2.2	63.6 ± 4.8	45.2 ± 3.5	36.4 ± 2.9
	(g) with $\beta = 0.5$	88.4 ± 2.0	69.2 ± 2.7	43.2 ± 3.7	28.0 ± 3.8
	(g) with $\beta = 2$	87.6 ± 3.2	72.4 ± 4.1	42.8 ± 4.1	35.2 ± 3.5
	(g) with $\beta = 4$	86.8 ± 2.0	65.6 ± 2.3	48.4 ± 5.0	38.4 ± 3.4
	(g) with $r_{\mathrm{TD}} = 4$	82.8 ± 2.0	64.0 ± 1.3	48.4 ± 6.1	35.2 ± 2.7
	(g) with $r_{\mathrm{TD}} = 12$	89.6 ± 2.0	68.0 ± 3.8	48.0 ± 4.0	32.4 ± 6.1
	(g) with $r_{\mathrm{TD}} = 16$	91.2 ± 2.4	64.4 ± 6.6	47.6 ± 3.4	36.8 ± 4.5

B.2 CUBE

Table 7 repeats the rows of Table 2 on Cube. Tasks, seeds and protocols match Table 1. The graph, subgoal and final-switch thresholds use Cube's own calibration ($H_{\mathrm{TD}} = 7.25$, lookahead and switch radius $u_{25} = 13.63$). Row (g) is the Cube row of Table 1.

Table 7: Cube ablation (success %, 50 tasks, five seeds, mean ± sd). Rows as in Table 2. **Bold**: best mean per column; underlined: ±sd interval overlaps the best's.

	configuration	same-ep 25	same-ep 50	same-ep 100	cross-ep
	L2 (LeWM)	56.0 ± 3.8	34.4 ± 3.2	35.6 ± 1.5	16.4 ± 1.5
(a)	TDR terminal cost (no graph)	50.0 ± 1.8	48.8 ± 2.0	50.4 ± 1.5	**36.8** ± 4.3
(b)	graph subgoal	49.2 ± 3.0	46.8 ± 5.0	46.0 ± 3.6	28.4 ± 4.1
(c)	(b) + local switch	54.4 ± 2.9	44.8 ± 2.7	38.4 ± 3.7	28.4 ± 4.3
(d)	L2 + ET critic (no graph)	**64.0** ± 4.4	45.2 ± 2.4	47.6 ± 3.7	17.6 ± 3.4
(e)	(b) + ET critic (no switch)	58.0 ± 1.8	46.8 ± 5.2	**56.8** ± 5.2	28.4 ± 6.1
(f)	(c) + $-\log V$ critic	56.8 ± 3.5	48.0 ± 2.2	39.6 ± 2.7	32.4 ± 4.8
(g)	(c) + ET critic (**ours**)	62.0 ± 4.7	**49.6** ± 4.6	50.4 ± 3.2	27.6 ± 6.1
(h)	(g), subgoal scored by L2	63.6 ± 4.6	42.4 ± 2.9	47.2 ± 1.6	26.4 ± 4.6

The ordering of components differs from Push-T. On Cube the TDR terminal cost alone (a) is the strongest configuration on cross-episode goals (36.8%), is within 0.8 points of the full method at same-ep 50 (48.8 vs. 49.6%), and ties it at same-ep 100 (50.4%). Adding the graph subgoal (b) does not improve on (a) in any protocol, and lowers cross-episode success by 8.4 points. The local switch (c) recovers 5.2 points at same-ep 25 but costs 7.6 points at same-ep 100. The ET critic is most useful at short range: on top of L2 alone (d) it adds 8.0 points at same-ep 25 and 10.8 at same-ep 50, but only 1.2 on cross-episode goals, whereas on Push-T the same row doubled cross-episode success. With the graph in place the critic changes cross-episode success by at most 4 points in either direction ((e)–(g) against (b)). The local switch also behaves differently: on Push-T it helps at every protocol, whereas on Cube removing it (e) gives the best result at same-ep 100 (56.8%).

On Cube, most of the full method's gain over L2 comes from the learned temporal distance. The TDR terminal cost alone (a) accounts for nearly all of the gain at same-ep 50 and 100 (+14.4 and +14.8 points over L2, against +15.2 and +14.8 for the full method) and exceeds it on cross-episode goals. The critic supplies the gain at same-ep 25, where TDR alone falls below L2. The graph route and the local switch, which drive the Push-T results, add little on Cube. The contribution of each component is therefore environment-dependent, and the component-wise conclusions of Section 5.2 are specific to Push-T. The combined method is nonetheless the most robust configuration: it is the only row that is best or within one standard deviation of the best on every protocol in both Table 2 and Table 7.

C Demonstration retrieval

Method. CEM's first population is sampled from $\mathcal{N}(\mathbf{0}, I)$ in z-scored action space, so every plan starts from "do nothing". Retrieval uses the dataset as an action library to choose a better starting mean. Let $\mathcal{T}$ be the training frames whose episode has at least $L = 25$ steps left (five 5-action blocks, one planning horizon). At each planning call with current latent z_0 and target t, where $t = c_{v_\lambda}$ while the graph subgoal is active and $t = \psi(z_g)$ after the local switch (Equation (4)), we select

$$i^\star = \arg\min_{i \in \mathcal{T}} \ \|\psi(z_i) - \psi(z_0)\| + \|\psi(z_{i+L}) - t\|, \tag{13}$$

by exact nearest-neighbour search on the GPU. The logged actions $a_{i^\star}, \ldots, a_{i^\star+L-1}$ are z-scored with the same normaliser the predictor uses, reshaped into five action blocks, and passed to CEM as its initial mean. The variance, number of samples, elites, iterations, cost and replanning interval are unchanged.

Motivation. Cube's action space is five-dimensional, against two dimensions on Push-T and Reacher. With a fixed budget of 300 samples per iteration, a zero-mean CEM population covers this larger space far more sparsely, so good action sequences are less likely to be sampled at all. Retrieval addresses this by starting the search from a demonstrated action block instead of from zero. We evaluate it on all five seeds and all four protocols, for all three environments (Table 4); the retrieval flag is the only difference from our method's runs.

Results. On Cube, retrieval wins in every protocol and on every seed at same-ep 25, same-ep 100 and cross-episode. The gain is smallest at same-ep 50 (+9.6) and largest at same-ep 100 (+28.4) and cross-episode (+20.4). Its seed-to-seed sd is lower than our method's in every protocol (e.g. 2.8 vs. 6.1 on cross-episode). On Push-T it loses 1.6–3.2 points in the same-episode protocols and ties on cross-episode, with higher variance. On Reacher it loses in every protocol, most at same-ep 25 (-14.0) and least on cross-episode (-2.0). Retrieval therefore does not help on Push-T or Reacher, likely because in their two-dimensional action spaces the CEM population alone already covers the space adequately, so a demonstrated starting mean adds little and can bias the search.

D LIVE RENDERING MUST MATCH THE DATASET

The frozen encoder is sensitive to renderer versions. Reacher frames rendered with MuJoCo 3.12 lack the floor checker texture of the dataset's MuJoCo 3.5 frames; the pixel difference is 18/255 on average and moves the latent by a median of 15 units, about one 25-step gap, so the terminal cost never reaches zero even with the arm at the goal. This alone took LeWM's own planner from 87% to 65% at offset 25.

E WHERE THE LATENT-L2 OBJECTIVE FAILS

This section traces one failure mode of terminal latent L2. Appendix E.1 states two properties of high-dimensional isotropic Gaussians that the rest of the section uses. Appendix E.2 measures how far apart start and goal latents sit as the horizon grows, and Appendix E.3 shows why a straight line between two such latents leaves the region the encoder populates. Appendix F checks the same failure inside CEM's own candidate ranking.

E.1 CHARACTERISTICS OF HIGH-DIMENSIONAL GAUSSIANS

Let $z \sim \mathcal{N}(\mu, \sigma^2 I_d)$, let χ_d denote the Euclidean norm of a standard Gaussian vector in $\mathbb{R}^d$, and write $m_d = \mathbb{E}[\chi_d] = \sqrt{2}\,\Gamma(\frac{d+1}{2})/\Gamma(\frac{d}{2})$ for the mean of a chi distribution with d degrees of freedom. By Wendel's inequality for ratios of Gamma functions (Wendel, 1948),

$$\frac{d}{\sqrt{d+1}} \;\le\; m_d \;\le\; \sqrt{d}. \tag{14}$$

Lemma 4 (Distance between random pairs)**.** *Let z_1, z_2 be independent draws of $\mathcal{N}(\mu, \sigma^2 I_d)$. Then $\|z_1 - z_2\| \sim \sqrt{2}\,\sigma\,\chi_d$, so*

$$\mathbb{E}\|z_1 - z_2\| = \sqrt{2}\,\sigma m_d \in \Big[\sqrt{2}\,\sigma\tfrac{d}{\sqrt{d+1}},\ \sqrt{2d}\,\sigma\Big], \qquad \mathrm{Var}\,\|z_1 - z_2\| = 2\sigma^2(d - m_d^2) \le 2\sigma^2.$$

In particular, $\mathbb{E}\|z_1 - z_2\| = \sqrt{2}\,\mathbb{E}\|z - \mu\|$.

Proof. $z_1 - z_2$ is Gaussian with mean 0 and covariance $2\sigma^2 I_d$, so $\|z_1 - z_2\|/(\sqrt{2}\sigma)$ is the norm of a standard normal vector in $\mathbb{R}^d$, which is χ_d-distributed. The mean follows from the definition of m_d and Equation (14). For the variance, $\mathbb{E}[\chi_d^2] = d$, so $\mathrm{Var}[\chi_d] = d - m_d^2 \le d - d^2/(d+1) = d/(d+1) < 1$ by the lower bound in Equation (14). The last claim holds because $\|z - \mu\| \sim \sigma\chi_d$. □

The typical distance between unrelated points therefore grows as $\sigma\sqrt{2d}$, while its standard deviation stays below $\sqrt{2}\,\sigma$. At $d = 192$ the ratio of standard deviation to mean is at most $\sqrt{d+1}/d < 7.3\%$: nearly all unrelated pairs are about equally far apart.

Proposition 5 (Gaussian shell)**.** *Let $r = \|z - \mu\|$ with $z \sim \mathcal{N}(\mu, \sigma^2 I_d)$. Then $\mathbb{E}[r] = \sigma m_d \in [\sigma d/\sqrt{d+1},\ \sigma\sqrt{d}]$ and $\mathrm{Var}[r] \le \sigma^2$. Moreover, for every $\varepsilon \in (0, 1]$,*

$$\Pr\big[r \le (1-\varepsilon)\,\sigma\sqrt{d}\,\big] \;\le\; \exp\!\big(-d\varepsilon^2/4\big).$$

Proof. $r/\sigma \sim \chi_d$, so the moments follow as in Lemma 4. For the tail, the lower-tail bound of Laurent & Massart (2000) gives $\Pr[\chi_d^2 \le d - 2\sqrt{dx}] \le e^{-x}$ for all $x > 0$. The event $r \le (1-\varepsilon)\sigma\sqrt{d}$ is the event $\chi_d^2 \le d(1-\varepsilon)^2$. Choose x with $d - 2\sqrt{dx} = d(1-\varepsilon)^2$, that is $\sqrt{x} = \sqrt{d}\,(2\varepsilon - \varepsilon^2)/2$. Since $2\varepsilon - \varepsilon^2 \ge \varepsilon$ on $(0, 1]$, $x \ge d\varepsilon^2/4$, and the bound follows. □

Proposition 5 says that the mass of a high-dimensional isotropic Gaussian lies in a thin shell of radius about $\sigma\sqrt{d}$ and width $O(\sigma)$, and that the interior of the ball is exponentially unlikely. One consequence matters for planning. If z_1, z_2 are independent, their midpoint satisfies $\frac{1}{2}(z_1 + z_2) - \mu \sim \mathcal{N}(0, \frac{\sigma^2}{2} I_d)$, so by Proposition 5 it concentrates at radius about $\sigma\sqrt{d/2}$, a factor $1/\sqrt{2}$ inside the shell. A straight line between two typical points therefore passes through a region that the distribution itself almost never visits.

E.2 Predictor rollout error by horizon

Rolling the frozen predictor forward under real logged actions and comparing to the true encoded endpoint, Push-T's latent rollout error grows with horizon but stays well below the scale of two unrelated frames (mean 19.6) even at $H = 100$ (Table 8 and Figure 3): the predictor degrades gracefully rather than collapsing to an uninformative guess. The random-pair scale agrees with Lemma 4: with the measured mean radius 13.88 (Appendix E.3), the lemma predicts a mean random-pair distance of $\sqrt{2} \times 13.88 \approx 19.63$, against the measured 19.6.

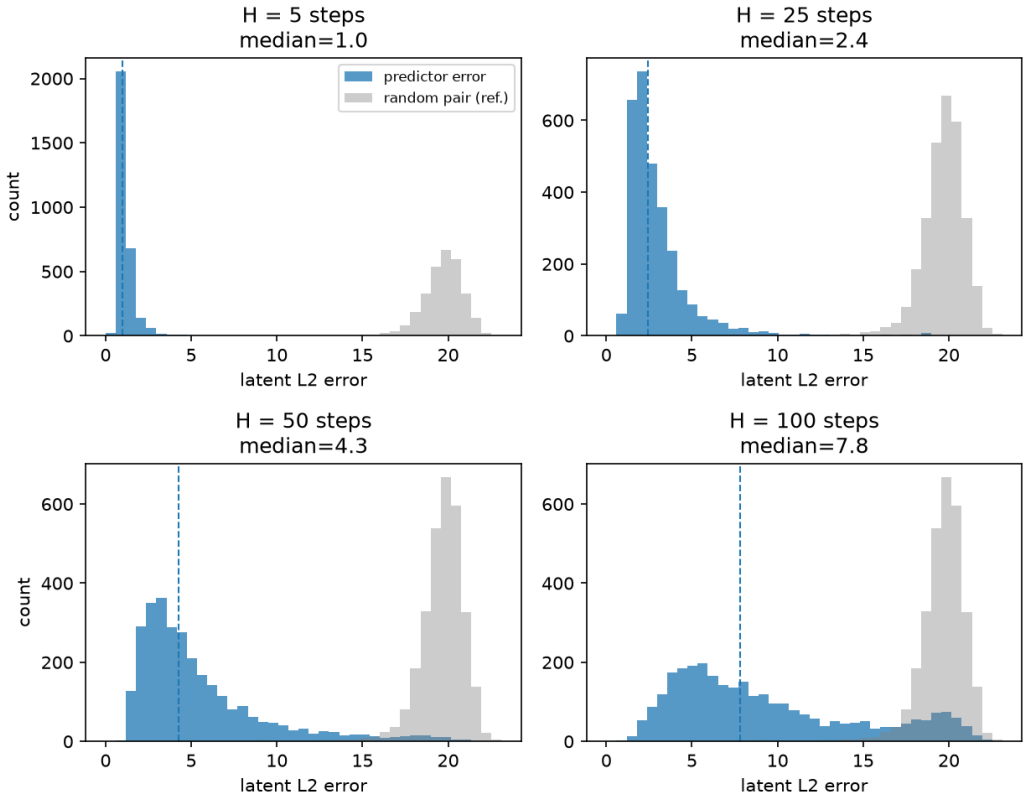


Figure 3: Push-T predictor latent rollout error (blue) vs. horizon, against the random-pair reference distribution (gray).

Task difficulty: how far apart are the actual start/goal pairs? The random-pair reference above (mean 19.6) is the scale at which two frames carry no information about each other. Sampling same-episode (start, start+H) pairs the same way Figure 3 samples its rollout starts, but reading off the true encoded distance $\|z_{\text{start}} - z_{\text{start}+H}\|$ instead of rolling the predictor forward (Table 8), shows that $H = 50$/$H = 100$ are close to indistinguishable from random-pairs (19.4, 19.6) – consistent with Section 5.3's Spearman collapse past H=50. Relative to that task scale, the predictor's own rollout error is at 14.1% of the gap at $H = 25$ and grows to 39.8% by $H = 100$ – a sizeable fraction

of the very distance CEM is trying to optimize over, consistent with L2's success falling as its plans reach further ahead (Table 9).

Table 8: Push-T predictor error compared with task distance. For each horizon H we report the latent L2 error of an H-step predictor rollout and the true distance between a start frame and the frame H steps later in the same episode (median $\pm$ std,). Relative error is the ratio of the two medians. For reference, two random frames lie 19.6 apart on average, matching the Gaussian prediction (Appendix E.3).

	$H = 5$	25	50	100
predictor error	1.00 ± 0.56	2.41 ± 2.09	4.28 ± 3.82	7.80 ± 5.35
start→goal distance	5.29 ± 2.76	17.06 ± 4.19	19.45 ± 3.29	19.60 ± 2.43
relative error (%)	18.9	14.1	22.0	39.8

A related question concerns the planner itself: how far ahead should the L2 baseline plan before replanning? Table 9 varies the plan length over 5 steps (a single action block), 25 steps (the default), 50 and 100 steps. The CEM horizon is set equal to the executed chunk in every case, so the planner executes exactly the span it evaluated. Budgets, tasks and seeds follow Table 1.

Table 9: Push-T L2 baseline versus plan length. The default is 25 steps. Success (%), mean $\pm$ sd over 5 CEM seeds

	plan length (env steps)			
protocol (budget)	5	25 (default)	50	100
same-ep 25 (50)	64.0 ± 1.8	**91.2** ± 1.0	–	–
same-ep 50 (100)	19.2 ± 2.7	**40.8** ± 3.7	34.4 ± 2.7	–
same-ep 100 (200)	1.2 ± 1.6	**13.2** ± 3.2	12.8 ± 2.4	5.6 ± 2.3
cross-ep (250)	4.4 ± 1.5	14.4 ± 3.9	**15.2** ± 4.8	4.8 ± 2.0

Success is non-monotonic in plan length, with the best results at 25–50 steps. Committing to 100 steps reduces L2's success by 7.6 points at same-ep 100 and 9.6 points at cross-ep, which is consistent with predictor error compounding over a long open-loop chunk. Shorter plans perform worse still. With a single-block lookahead the planner becomes greedy and cannot plan the multi-step approach required before the block moves. At long range, the 50-step setting is within seed variation of the default. The default 25-step plan length is at or near the best setting for L2, and the gaps reported in Table 1 do not arise from the choice of plan lengths.

E.3 Why a straight latent line goes out of distribution

Appendix E.2 shows the predictor itself degrades gracefully with horizon; this section asks why the terminal L2 objective built on top of it still fails past $H \approx 50$ (elite-ranking Spearman 0.396 → 0.086, Appendix F). By Proposition 5, the mass of a high-dimensional isotropic Gaussian lies on a thin shell of radius about $\sigma\sqrt{d}$. If LeWM's Push-T latent cloud approximates such a shell, a straight line drawn between two points *on* the shell passes through the ball's interior, which the shell distribution assigns almost no mass – exactly the regime a z-space-L2 CEM plan implicitly cuts through when it scores a candidate purely by terminal distance to a goal many steps away.

The latent cloud is close to an isotropic shell. LeWM's SIGReg term regularises the encoder towards an isotropic Gaussian, so the theory above should apply to its latents directly. We test this on encoded Push-T frames ($d = 192$). The latents are centred: their empirical mean has norm 0.59, against a typical latent norm of 13.9, so we take $\mu = 0$. With $\sigma^2 = \mathrm{tr}(\Sigma)/d$ the average per-dimension variance, the theory predicts that $\|z\|$ follows $\sigma\chi_d$, with mean 13.88. The encoded frames match this almost exactly (empirical mean 13.89, Figure 4).

A straight-line midpoint is far off the shell; the real trajectory is not. For 200,000 same-episode frame pairs 100 steps apart (the same-ep-100 horizon), both endpoints sit on the shell as expected (start mean 14.15, goal mean 13.60), and so does the actual logged frame at the trajectory's

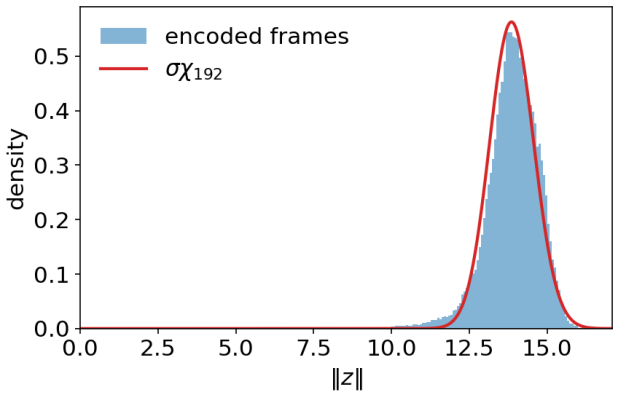


Figure 4: Push-T: norm $\|z\|$ of encoded frames (histogram of 200,000 of the 2.3M frames) vs. the $\sigma\chi_{192}$ distribution predicted for a zero-mean isotropic Gaussian with the same per-dimension variance ($\sigma = 1.003$). Empirical mean 13.89 (sd 0.82); predicted mean 13.88 (sd 0.71).

true midpoint (mean 13.81). The linear interpolation $\frac{1}{2}(z_{\text{start}} + z_{\text{goal}})$ does not: its mean radius is 10.03, 5.42 shell-σ below the shell mean, and 96.4% of the 200,000 interpolated midpoints fall below the shell's own 1st percentile radius (Figure 2 in Section 5.3). The two populations do not overlap. This isolates the effect to interpolation itself, not to trajectory midpoints being unusual in general: the real dynamics stay on the shell, curving around it rather than cutting through its interior, which is exactly the structure a graph built in a temporal-distance representation is meant to follow instead of a raw latent straight line (Sections 3 and 5.2).

F Candidate ranking within CEM's own elites

The ranking comparison in Section 5.3 is performed on candidates drawn from the planner's own search rather than on a fixed diagnostic pool. This appendix defines the procedure.

Candidate set. For each task i, with start state $s_0^{(i)}$ and goal observation $o_g^{(i)}$, we run the first CEM solve of OUR and stop after iteration 1. At this iteration both costs are evaluated on the same population of $N = 300$ action sequences, sampled from CEM's default initialisation $\mathcal{N}(\mathbf{0}, I)$ in normalised action space. No elite selection has yet shaped this population, so it is unbiased with respect to either cost. Let $\mathcal{E}_i = \{a^{(i,1)}, \ldots, a^{(i,30)}\}$ be the 30 elites selected by OUR's cost. Each $a^{(i,k)}$ is a 25-step sequence of raw actions, obtained by de-normalising and clipping to the action bounds.

Costs. For every $a \in \mathcal{E}_i$, the frozen predictor produces a single imagined rollout. We evaluate two costs on this rollout: OUR's planning cost $c_{\text{OUR}}(a)$ (the complete objective of Section 3), and LeWM's terminal cost $c_{\text{L2}}(a) = \|\hat{z}_K(a) - z_g\|_2^2$. The two costs therefore differ only in the criterion, not in the predicted trajectory.

Oracle label. Each elite is executed in the simulator from $s_0^{(i)}$, reaching the true state $s^{(i,k)}$. From $s^{(i,k)}$, a privileged CEM planner with access to the true simulator (population 64, 5 iterations, 2 independent trials) searches for the smallest number of steps after which the environment's success predicate holds for goal $o_g^{(i)}$. The recovery time is

$$T^{(i,k)} = \min\{\, t \le T_{\max} : \text{the oracle reaches the goal in } t \text{ steps} \,\}, \tag{15}$$

where the minimum is taken over trials, and $T^{(i,k)} = T_{\max} + 1$ if no trial succeeds within the cap $T_{\max} = 100$. Such labels are right-censored: they indicate that no plan was found, not that recovery is impossible. Lower T indicates a better candidate. The oracle uses the simulator state and is not available to either planner.

Statistic. For each task we compute the Spearman rank correlation $\rho_i(c) = \mathrm{Spearman}\big(\{c(a^{(i,k)})\}_{k=1}^{K}, \{T^{(i,k)}\}_{k=1}^{K}\big)$ for $c \in \{c_{\mathrm{OUR}}, c_{\mathrm{L2}}\}$. A positive value means that the cost assigns lower values to candidates that recover faster. Tasks on which all K labels are equal (typically all censored) have undefined ρ_i and are excluded. We report the mean $\bar{\rho}(c) = |\mathcal{I}|^{-1} \sum_{i \in \mathcal{I}} \rho_i(c)$ over the set $\mathcal{I}$ of remaining tasks, with 95% confidence intervals from a bootstrap over tasks.

Setup. We use 25 tasks from each task pool, CEM seed 0, and learned-asset seed 0. Because $\mathcal{E}_i$ is selected by c_{OUR}, the comparison measures how well each cost orders the candidates that OUR would retain. It does not measure how well each cost ranks the full population.

Table 10: Mean per-task Spearman correlation $\bar{\rho}$ between each cost and the oracle recovery time T over OUR's 30 elites at CEM iteration 1. $|\mathcal{I}|$ is the number of tasks with non-degenerate labels. At same-ep 100, 14 of 25 tasks were fully censored even with a larger oracle cap.

protocol	$\lvert\mathcal{I}\rvert$	$\bar{\rho}(c_{\mathrm{OUR}})$	$\bar{\rho}(c_{\mathrm{L2}})$
same-ep 25	25	0.327	0.396
same-ep 50	24	0.203	0.086
same-ep 100	11	0.191	−0.064

Result. At same-ep 25, c_{L2} has the higher mean correlation. At same-ep 50 and 100, c_{OUR} has the higher correlation. The L2 ranking at these horizons is not distinguishable from chance. The crossover matches the short-range/long-range pattern of closed-loop success in Table 1, here measured on the candidate ranking of a single CEM iteration.

G Parameter counts

Table 11 lists the parameters of the frozen LeWM stack and of the two components we train. Only ψ_0 of the TDR's two networks is used at planning time.

Table 11: Push-T parameter counts. All LeWM modules are frozen.

LeWM encoder (ViT-tiny)	5.50M
LeWM predictor	10.79M
LeWM action encoder	0.16M
LeWM projector (encoder head)	0.80M
LeWM predictor projector	0.80M
LeWM total (frozen)	18.04M
TDR (used, ψ_0 only)	0.64M
TDR (full 2-net ensemble)	1.29M
Critic (per member)	0.83M
Critic (5-member ensemble)	4.13M

H Hyperparameters

Table 12 lists hyperparameters shared across Push-T, Reacher, and Cube; Table 13 lists ones that are environment specific. The per environment scalars in Table 13 are calibrated directly from the statistics inferred from the dataset, and arent tuned per environment. CEM uses LeWM's released evaluation configuration in every environment; the LeWM paper reports 10 iterations outside Push-T.

Table 12: Hyperparameters shared across all three environments. TDR and graph follow Section 3.1; critic follows Section 3.3; CEM and the criterion follow Equation (7) and Algorithm 1.

component	hyperparameter	value
TDR ψ	architecture	3×512 LayerNorm+GELU MLP, linear head
	embedding dim	32
	goal relabelling	$\mathrm{Geom}(1-\gamma)$ future w.p. 0.625, else random
	value networks	2 (TD target: min)
	expectile τ	0.99
	discount γ	0.99
	training steps	50,000
	batch size	1024
	learning rate	3×10^{-4}
Graph	TE threshold θ_{TE}	0.9
	clustering radius	$r_{\mathrm{TD}}/2$
	goal attachment radius	$\max(r_{\mathrm{TD}},\ 1.2\,d_{\min})$
Critic V	ensemble members	5
	architecture (per member)	3×512 MLP + SiLU, linear head
	max horizon $h_{\max}$	50 env steps
	horizon grid	multiples of 5, 0–$h_{\max}$
	training steps	60,000
	hindsight radius for $k^\star$	u_2
	Bellman candidates M	8 (1 logged, 4 perturbed, 3 random)
	Bellman pessimism κ	1.0
	loss weights ($\lambda_{\mathrm{Bellman}}$, λ_{mono})	1.0, 1.0
	learning rate	3×10^{-4}
	EMA target decay	0.995
	cross-episode negative gate	$d_\psi(z, z_g) \geq 1.25\,u_{\max(h,5)}$
CEM (LeWM, unchanged)	population / iterations / elites	300 / 30 / 30
	horizon K	5 blocks ⇒ 25 env steps
	replan interval (receding horizon)	25 env steps
Criterion (Equation (7))	critic weight β	1
	critic cost	ET (budget-capped expected hitting time)
	metric after the local switch	LeWM's own z-space L2 to the goal

Table 13: Environment-specific hyperparameters. u_k denotes the calibrated TDR distance of a k-env-step transition (median over training transitions), read from each environment's own TDR calibration. $\lambda = u_{25}$ and $r_{\mathrm{TD}} = u_{12}$ in every environment. Every other hyperparameter is identical across environments.

	Push-T	Reacher	Cube
r_{TD} (TDR units)	7.59	3.72	7.25
$\lambda = u_{25}$ (TDR units)	13.53	5.83	13.63
TE lag k_{TE} (env steps; r_{TD} rounded)	8	4	7
graph nodes	20.5k	535	16k

I Detailed theory and proofs

I.1 Scope, state, and time conventions

We hold the trained encoder, predictor, TDR, graph, and critic fixed throughout the analysis. We examine whether the controller reduces the remaining graph distance, whether this progress brings it to a region where local control reaches the goal, and when the critic's score measures an expected time to reach the goal. Graph connectivity alone does not establish these properties. All error tolerances and optimization gaps below are nonnegative; the graph radius, lookahead, and normalization floors are positive.

Let x denote a physical state, or a history state sufficient to make the dynamics Markov. Write $z(x)$ for its current encoded observation and $\psi(x) = \psi(z(x))$ when no confusion results. The LeWM predictor uses its input history as well as its current latent. In the deterministic results, $x^+(u)$ is the state after the entire sequence u of $L = K\Delta$ primitive actions, and $z^+(u) = z(x^+(u))$. Write $\hat{z}^+(u)$ for the predicted endpoint. We bound prediction error for the observed input history. The graph-descent theorem therefore does not require a single image latent to be Markov. The Bellman results require a state that contains enough information to be Markov. If the implemented critic receives only $z(x)$, its error must include information lost by that input restriction.

Fix a goal observation z_g. Let $\mathcal{G}_g$ be the measurable set of environment states that satisfy the task's goal condition. Raw latent distance to z_g need not identify this set. For example, a goal may constrain only an object's position while images also contain an unconstrained robot configuration. Conversely, two identical rendered arm poses can have different unwrapped joint coordinates. The local theorem explicitly assumes that a small enough latent error implies task success. Matching goal images alone does not ensure that every environment's goal condition holds. When the agent satisfies the goal condition, the controller ends the episode and skips any remaining actions in the current plan. We need endpoint bounds only for calls that have not already reached the goal. If the agent reaches the goal during a call, it has already satisfied the theorem's conclusion, even if its latent lies outside the TDR switching region. The graph phase therefore ends when the agent first reaches the goal or enters $\mathcal{R}_\lambda$.

Definition 6 (Block hitting time)**.** Under continuation policy π, let τ^π be the first environment step at which the agent enters $\mathcal{G}_g$, with value ∞ if it never enters. Define $T^\pi = \Delta\lceil \tau^\pi/\Delta \rceil$, including $T^\pi = 0$ for an initially successful state. The block process records whether success occurred at any time within a block and makes that event absorbing.

We count the first time the agent satisfies the goal condition as success; the physical object need not remain at the goal afterward. A block transition that inspects only its final image may miss an earlier success. Such an omission is a modeling error when the intended labels record first hits. The state in the Bellman analysis includes the hit flag when it is needed.

All empirical budgets are in primitive environment steps. The critic grid has spacing $\Delta = 5$, the MPC horizon is $K = 5$ blocks, and the controller executes up to $L = 25$ steps per call, stopping early if it reaches the goal. The main protocols have budgets divisible by L. If the remaining budget is smaller than L, the controller must shorten the plan and update the clock by the number of executed steps, or stop. The full-L formulas apply only when at least L steps remain. We learn the discounted TDR from consecutive-frame transitions; it has its own calibrated units. Neither r_{TD} nor λ is itself a count of environment steps.

Relationship to earlier guarantees. The representation theorem in HILP links embedding and directional-action error to optimal goal-reaching under its stated deterministic assumptions (Park et al., 2024). Our controller instead tracks a node chosen by graph search and executes a whole action sequence. The proof below works with the finite graph's actual path length, without assuming a globally accurate embedding of optimal temporal distance. PcLast and ALPS use learned representations for hierarchical model-based planning (Koul et al., 2024; Shehmar et al., 2026); the spectral commute-time identity in ALPS concerns representation geometry, whereas we bound progress under the controller's combined cost. The terminal proof uses the local contraction structure familiar from MPC (Mayne et al., 2000). The reachability recursion is a finite-horizon dynamic program (Abate et al., 2008), which uses different assumptions from infinite-horizon stochastic shortest-path convergence (Bertsekas & Tsitsiklis, 1991).

Correspondence between the algorithm and the assumptions. The potential in Equation (8) is the start-attachment objective in Equation (3), using the stored TDR centers and the same fixed goal graph. The path length includes the start attachment, as the algorithm's lookahead rule requires. The comparison sequence contains K action blocks. We compare the frozen predictor's endpoint with the endpoint after executing all L steps. This assumption requires a suitable action sequence, without requiring a learned low-level policy. We score CEM's returned mean and the comparison sequence using the final population's normalization constants. We do not recompute the IQR after adding either sequence. Earlier CEM iterations may use different constants; the proof compares both sequences under the same final objective.

The critic remains in the cost after the local switch, so differences in its scores contribute to both Equations (9) and (11). The Bellman reference allows feedback between Δ-step blocks, whereas the controller commits to K such blocks; an accurate reference critic is therefore not automatically the value of this controller. The common-policy survival interpretation requires a separately specified continuation policy. Finally, the tabular expectile theorem analyzes exact value-table updates. TDR training instead uses twin networks, lagged targets, and Euclidean distances. The main progress theorem treats the trained representation as fixed and does not require its optimizer to converge.

If a variant scores the mapped medoid $\tilde{y}_q = \psi(z_q)$ instead of the graph center y_q, let $\delta_q = \|\tilde{y}_q - y_q\|$. If the comparison sequence ends within η of $\tilde{y}_q$, the same argument bounds the executed endpoint's distance to the graph center by $\rho + \delta_q$; use that quantity in the radius and descent conditions. If η instead measures the comparison sequence's endpoint distance to y_q, use $\rho + 2\delta_q$. Both statements follow by one triangle inequality at each changed target comparison. Replacing centers by medoids therefore changes the error bound; the formulas must account for the distance between the two targets.

I.2 Restricted attachment and path progress

Let G_g be a finite graph augmented by the goal node, with coordinate $y_g = \psi(z_g)$. Its edges have nonnegative weights equal to Euclidean distances between their coordinates. $D_g[v]$ is finite only for nodes connected to the goal. Choose shortest paths without repeated nodes; such paths exist even with zero-weight edges. Define

$$A(z) = \{v \in V_G : \|\psi(z) - y_v\| \le r_{\mathrm{TD}},\ D_g[v] < \infty\}.$$

The goal is not needed as an extra member of $A(z)$: the explicit local switch handles states near it. If $A(z)$ is empty, $\Phi(z) = +\infty$, and no finite graph-progress bound applies. The following results assume that at least one attachment has a finite route to the goal.

Lemma 7 (Path prefix and attachment)**.** *Let $v^\star$ attain Equation* (8)*, and let q be a node on a selected shortest path from that attachment to the goal. Let p_q be the cumulative path length from $\psi(z)$ to y_q, including the initial attachment segment. Then*

$$\Phi(z) = p_q + D_g[q], \qquad d_\psi(z, z_g) \le \Phi(z). \tag{16}$$

Here $D_g[g] = 0$. If q is a regular graph node and $\|\psi(z') - y_q\| \le r_{\mathrm{TD}}$, then

$$\Phi(z') \le \Phi(z) - p_q + \|\psi(z') - y_q\|. \tag{17}$$

Proof. Every suffix of the selected shortest path is a shortest path from its initial node to the goal. Otherwise replacing that suffix with a shorter one would shorten the selected path. Its total length, including the attachment cost, is $\Phi(z)$, which proves the first identity. Applying the Euclidean triangle inequality successively to the coordinates along the same path proves the second inequality. If the displayed distance from z' to q is at most r_{TD}, the node q is a feasible member of $A(z')$: it has finite $D_g[q]$ because it lies on the route. Evaluating the minimum defining $\Phi(z')$ at q proves Equation (17). □

The proof does not assume that D_g has a Lipschitz extension or that Φ is continuous. The radius restriction can make Φ jump when a node enters or leaves the attachment set. We therefore require the next endpoint to lie within r_{TD} of the selected node. A qualitative statement that the endpoint is nearby would not suffice.

Lemma 8 (First-crossing lookahead)**.** *Suppose $z \notin \mathcal{R}_\lambda$. The selected path has length greater than λ and therefore contains a first λ-crossing target. If r_e is the largest segment on the augmented path, including the initial attachment, then*

$$\lambda \le p_q < \lambda + r_e.$$

One may take $r_e \le \max(r_{\mathrm{TD}}, r_g)$ for the stated graph construction.

Proof. By Lemma 7, $\Phi(z) \ge d_\psi(z, z_g) > \lambda$. The finite cumulative sequence starts at zero and ends at $\Phi(z)$. Just before its first crossing it is strictly below λ; the crossing adds one segment of length at most r_e. Regular edges and the initial attachment are at most r_{TD}, and goal-attachment edges are at most r_g. □

The selected target can therefore lie beyond the calibrated lookahead distance. In particular, expanding the goal-attachment radius can create a much longer last edge. The comparison sequence in Theorem 1 must reach near this actual target, including any extra distance from crossing the lookahead threshold. Likewise, resetting cluster centers to means after sequential admission does not preserve an admission-radius guarantee for all members. The proof needs the actual center distances and endpoint error, not an unproved bound on cluster radii.

Lemma 9 (Graph descent from endpoint tracking)**.** *Suppose a first-crossing target is tracked with error $\|\psi(z') - y_q\| \le \rho$, where $\rho \le r_{\mathrm{TD}}$ and $\rho < \lambda$. Then either $z' \in \mathcal{R}_\lambda$ or $\Phi(z') \le \Phi(z) - \lambda + \rho$.*

Proof. If q is regular and z' is outside $\mathcal{R}_\lambda$, apply Lemma 7 and $p_q \ge \lambda$. If $q = g$, the tracking error is $d_\psi(z', z_g) \le \rho < \lambda$, so $z' \in \mathcal{R}_\lambda$. These cases exhaust the possibilities. □

Why geometric edges require an execution assumption. An undirected radius graph can connect two latent clusters even if the physical dynamics permit travel in only one direction, or no travel at all. The temporal-efficiency filter concerns observed directions within episodes and does not prove reversibility of arbitrary inter-cluster edges. A finite D_g therefore does not ensure that the agent can follow the route. The comparison-sequence assumption requires an executable plan to the chosen target. It allows other graph edges to be infeasible, but requires more than proximity between the current state and target coordinates.

I.3 The normalized CEM cost and execution error

Fix one planning call, target, horizon grid, and set of normalization constants. Write

$$\widehat{b}(u) = \|\psi(\hat{z}^+(u)) - y_q\|, \quad \widehat{S}(u) = \Delta \sum_{j=0}^{n} [1 - V(\hat{z}^+(u), z_g, j\Delta)].$$

Since the critic outputs lie in $[0, 1]$, $0 \le \widehat{S}(u) \le C_n$ even if they are not monotone in the horizon. Denote the two population medians by m_b, m_S.

Lemma 10 (Exact scale-ratio comparison)**.** *If*

$$\frac{\widehat{b}(\hat{u}) - m_b}{s_b} + \beta \frac{\widehat{S}(\hat{u}) - m_S}{s_S} \le \frac{\widehat{b}(u^\circ) - m_b}{s_b} + \beta \frac{\widehat{S}(u^\circ) - m_S}{s_S} + \epsilon_{\mathrm{opt}},$$

with $s_b, s_S > 0$ and $\beta \ge 0$, then

$$\widehat{b}(\hat{u}) \le \widehat{b}(u^\circ) + s_b \epsilon_{\mathrm{opt}} + \beta \frac{s_b}{s_S} [\widehat{S}(u^\circ) - \widehat{S}(\hat{u})]_+. \tag{18}$$

Proof. Cancel the medians, move the critic difference to the right, and multiply by s_b. Replacing a negative critic difference by zero can only enlarge the right side. □

Normalization still leaves a tradeoff between the two costs. For example, multiplying raw critic scores by a positive constant generally cancels through their IQR when the floor is inactive, but a very small critic IQR can amplify small, unreliable variations. The numerical floors make the objective defined; they do not establish that the resulting scale ratio is appropriate. If two candidates use different population medians or scales, those constants do not cancel and the comparison above no longer applies.

Proof of Theorem 1. If the agent reaches the goal before the call ends, the conclusion already holds. Otherwise, the triangle inequality and the model-error assumption give

$$\widehat{b}(u^\circ) \le \|\psi(z^+(u^\circ)) - y_q\| + \|\psi(\hat{z}^+(u^\circ)) - \psi(z^+(u^\circ))\| \le \eta + e_\psi.$$

Apply Lemma 10, then bound the distance between the predicted and true endpoints of the returned sequence:

$$\begin{aligned} \|\psi(z^+(\hat{u})) - y_q\| &\le \widehat{b}(\hat{u}) + e_\psi \\ &\le \eta + 2e_\psi + s_b \epsilon_{\mathrm{opt}} + \beta(s_b/s_S)\omega = \rho. \end{aligned}$$

Lemma 9 proves the one-call claim. For the iteration bound, let τ_G be the first call that reaches the goal or has an endpoint in $\mathcal{R}_\lambda$, and let $a_G = \lambda - \rho > 0$. If $\tau_G > N$, all of the first N transitions remain outside $\mathcal{R}_\lambda$, so induction gives $0 \le \Phi(z_N) \le \Phi(z_0) - N a_G$. For $N = \lceil \Phi(z_0)/a_G \rceil$, the right side is nonpositive. Equality is also impossible outside $\mathcal{R}_\lambda$, since $\Phi(z_N) \ge d_\psi(z_N, z_g) > \lambda > 0$. Thus $\tau_G \le N$. If the initial state is already successful or in $\mathcal{R}_\lambda$, take $N_G = 0$. □

What the optimization assumption means for CEM. The comparison need not assume a global minimizer. It can use any feasible sequence u° whose tracking error is controlled, provided the returned sequence has the stated objective gap relative to that sequence. CEM's final mean is not necessarily a sampled candidate or an elite, and a mean of successful sequences can fail in a nonlinear system. Therefore a proof concerning only the best elite would not prove the theorem for the stated implementation. Here ϵ_{opt} compares the sequence the controller actually executes with the feasible comparison sequence, using the final population's fixed constants. Thirty CEM iterations do not establish that this gap is small (de Boer et al., 2005). One could evaluate the returned mean and retain a candidate with a lower score. The reported runs do not establish that they used this additional check.

Proposition 11 (Action coverage under adaptive sampling)**.** *Let $\mathcal{U}_\eta$ be sequences meeting a prescribed true tracking tolerance. Suppose every conditional CEM proposal, given all previous draws, places probability at least $p > 0$ on $\mathcal{U}_\eta$. Among N draws the probability of seeing no such sequence is at most $(1-p)^N$.*

Proof. Let F_i be the event that the first i proposals miss $\mathcal{U}_\eta$. Conditional on the past and F_{i-1}, the next failure probability is at most $1 - p$. The tower property yields $\mathbb{P}(F_i) \le (1-p)\mathbb{P}(F_{i-1})$. Induction proves the result; independence between iterations is not required. □

This result bounds the chance of sampling a useful sequence under the stated coverage assumption. If CEM concentrates its Gaussian proposals too narrowly, the lower bound p may fail. Even when CEM samples a good sequence, its final mean need not be good. Retrieval changes which actions CEM is likely to sample. It can therefore help independently of the cost, as the Cube results suggest.

Prediction error over one planning call. Suppose true and predicted histories admit a common normed state space with block transitions f and $\hat{f}$, the initial history error is zero, and

$$\|\hat{f}(x,a) - f(x,a)\| \le \varepsilon_P, \qquad \|f(x,a) - f(x',a)\| \le L_f \|x - x'\|$$

on all compared rollout states and actions. Applying both transitions to the same action sequence and using induction gives

$$e_K \le \varepsilon_P \sum_{i=0}^{K-1} L_f^i. \tag{19}$$

Indeed $e_{k+1} \le \varepsilon_P + L_f e_k$ by adding and subtracting the true transition from the predicted state. If the map from history to the endpoint TDR coordinate is L_ψ-Lipschitz, take $e_\psi \le L_\psi e_K$. The bound is $K\varepsilon_P$ for $L_f = 1$, is bounded uniformly in K for $L_f < 1$, and can grow geometrically for $L_f > 1$. These assumptions suffice to bound endpoint error. A small latent prediction loss alone does not imply them, especially if the encoded history cannot determine the next state. The graph supports progress across many short calls, while this model bound covers only one call. Model-based RL analyses also use short rollouts to limit accumulated error (Janner et al., 2019). Here we do not assume globally stable LeWM dynamics. Each replan starts from new observations, but the endpoint bound must still hold at every state the controller reaches.

I.4 Terminal-region convergence and the complete budget

The switched branch uses squared raw latent distance, while the graph branch uses an unsquared TDR distance. The distinction changes the optimization-error term from linear to square-root dependence. We include the critic in the local-phase proof because the controller still uses it.

Proof of Theorem 2. If a call succeeds before its endpoint, the success conclusion already holds. At a local state of error e, let u° be the contracting comparison sequence. Its true endpoint error is at

most ae, and its predicted endpoint error is at most $ae + e_z$. Lemma 10, now applied to the squared base cost, gives

$$\|\hat{z}^+(\hat{u}) - z_g\|^2 \le (ae + e_z)^2 + s_b\epsilon_{\text{opt}} + \beta(s_b/s_S)\omega_\ell.$$

All terms added on the right are nonnegative. Taking square roots and using $\sqrt{v^2 + w} \le v + \sqrt{w}$ for $v, w \ge 0$, followed by the endpoint model bound, yields

$$e(z^+(\hat{u})) \le ae + 2e_z + \sqrt{s_b\epsilon_{\text{opt}} + \beta(s_b/s_S)\omega_\ell} = ae + d_\ell.$$

Invariance ensures that the next call again uses this branch and its assumptions. Iteration gives $e(z_m) \le a^m e(z_0) + d_\ell \sum_{i=0}^{m-1} a^i \le a^m R + (1 - a^m)e_\infty$. It suffices that $a^m R \le r_{\text{goal}} - e_\infty$, which is precisely the stated conservative bound on m. If $R = 0$, entry is already successful. The assumption that latent error within r_{goal} implies task success proves the claim. Adding the graph and local call counts proves the execution-budget bound. Reaching the goal during either phase only shortens the execution. □

Ensuring that the controller stays in the local region. One sufficient special case is that, on a local domain, $\mathcal{R}_\lambda = \{z : e(z) \le R\}$ and $aR + d_\ell \le R$. The contraction inequality then leaves the state in $\mathcal{R}_\lambda$. More generally, each local call must end inside the TDR switching region until success. Small raw latent error alone does not ensure this; one must relate raw latent distance to TDR distance. If the implemented controller exits $\mathcal{R}_\lambda$, the local proof stops and the graph phase can resume. Neither theorem alone rules out repeated switching. The assumption that local calls stay in the region is therefore necessary for this proof.

Success tolerance and an unavoidable error floor. With persistent model or optimization error, the bound approaches e_∞ rather than zero. The strict condition $e_\infty < r_{\text{goal}}$ ensures that the error bound eventually enters the success tolerance. If $e_\infty = r_{\text{goal}}$ and R is larger, the upper bound approaches the boundary without reaching it in finite time. If unsuccessful states can lie arbitrarily close to the goal latent, no positive latent radius guarantees task success. This matters when the goal depends on state coordinates that the image does not reveal.

Clock dependence and available time. The ET grid and its IQR change with $B - t - L$. To combine Theorems 1 and 2, choose error constants that cover every call at its remaining budget. Values measured only at the initial budget need not suffice. If $B < L(N_G + N_\ell)$ the theorem does not assert success within the episode, although the controller may succeed faster than the bound. Increasing the number of steps executed before replanning requires a new model-error bound.

I.5 Stochastic execution and expected progress

We now allow individual calls to violate the deterministic assumptions with bounded probability. The first result bounds success probability within a fixed budget by bounding the chance of any such violation. The second bounds the expected number of graph calls and also requires a limit on how much a failed call can increase the remaining distance.

Corollary 12 (Finite-budget success event)**.** *Let $N = N_G + N_\ell$ bound the number of calls whenever all deterministic assumptions hold. Conditional on earlier calls satisfying their bounds and on their history, suppose call j obeys the appropriate one-call error and comparison bounds with probability at least $1 - p_j$. Include the bound $e \le R$ on local entry and retention in $\mathcal{R}_\lambda$ for local calls that do not succeed. Take $0 \le p_j \le 1$. If $B \ge LN$, success probability is at least $1 - \sum_{j=1}^{N} p_j$ (or zero when that lower bound is negative).*

Proof. The probability that the first violated call occurs at j is at most p_j, by conditioning on the history up to that call. These first-failure events are disjoint. On their complement the deterministic bound applies for all required calls. Summing their probabilities proves the lower bound. Once the agent reaches the goal, no later call can fail because the episode has ended. The proof does not require independence between calls. □

An ensemble standard deviation alone does not give the p_j values. To apply the corollary, one must justify the conditional execution or model-error probabilities separately; the corollary then combines them.

Proposition 13 (Expected graph-phase duration)**.** *Let τ_G be the first call with success or local-region entry, and set $W_j = \Phi(z_j)$ before τ_G and $W_j = 0$ afterward, for a fixed initial state. Assume these stopped potentials are finite and integrable. Before τ_G, suppose a good call decreases W by at least $a_G > 0$ with conditional probability at least $1 - p$, while any other call increases it by at most M. Here $0 \le p \le 1$ and $M \ge 0$. If $\delta = (1-p)a_G - pM > 0$, then τ_G satisfies*

$$\mathbb{E}[\tau_G] \le W_0/\delta.$$

Proof. For $j < \tau_G$, the conditional expected increment is at most $-(1-p)a_G + pM = -\delta$. After stopping the increment is zero. Hence

$$\mathbb{E}W_{(n+1)\wedge\tau_G} \le \mathbb{E}W_{n\wedge\tau_G} - \delta\,\mathbb{P}(\tau_G > n).$$

Sum for $n = 0, \dots, m-1$ and use nonnegativity of the left-hand potential: $\delta \sum_{n=0}^{m-1} \mathbb{P}(\tau_G > n) \le W_0$. Monotone convergence and the integer tail-sum identity give the result. □

For the deterministic progress bound, $a_G = \lambda - \rho \le \lambda$. Reaching the goal or entering $\mathcal{R}_\lambda$ also gives the required decrease: outside that set $\Phi(z) > \lambda \ge a_G$, so resetting W to zero decreases it by at least a_G. If a failed call can lose all finite graph attachments, no finite M follows. This result therefore does not cover calls that disconnect the agent from the goal graph.

I.6 What the survival score measures

For a specified policy π, write $F_j^\pi(x) = \mathbb{P}_x^\pi(T^\pi \le j\Delta)$ and $S_n^\pi(x) = \Delta \sum_{j=0}^{n}(1 - F_j^\pi(x))$. The probabilities are nondecreasing in j because the underlying first-hit events are nested.

Proposition 14 (Survival identity, truncation, and ranking)**.** *For any such policy, including one with a positive probability of never hitting,*

$$S_n^\pi(x) = \mathbb{E}_x^\pi[\min(T^\pi, C_n)], \qquad C_n = (n+1)\Delta. \tag{20}$$

If $F_j^a \ge F_j^b$ for every queried j, then $S_n^a \le S_n^b$. If one inequality is strict, the score inequality is strict. Two states with the same final-deadline probability can nevertheless have different S_n.

Proof. For every $T \in \{0, \Delta, 2\Delta, \dots, \infty\}$, the pointwise identity

$$\min(T, (n+1)\Delta) = \Delta \sum_{j=0}^{n} \mathbf{1}_{\{T > j\Delta\}}$$

counts the grid intervals before truncation. Because the sum is finite, we can take expectations term by term even when $\mathbb{E}T$ is infinite. This proves the first claim. Subtracting the sums proves the dominance statements. For the last claim, take deterministic times Δ and 2Δ with $n \ge 2$: both final probabilities are one but their scores are Δ and 2Δ. □

The endpoint convention. The grid $j = 0, \dots, n$ includes a term at the final queried deadline $n\Delta$. Truncating the hitting time at that deadline would require summing only $j = 0, \dots, n-1$, giving zero when $n = 0$. Our score retains the last term and therefore has cap C_n. At zero remaining time, it is $\Delta(1 - V_0)$, an endpoint-success penalty. At the maximum query horizon of 50 steps it has cap 55 steps. This extra interval changes rankings whenever V_n differs across candidates; so removing it can change which candidate the planner prefers.

If τ^π is the primitive hitting time, rounding introduces a bounded discrepancy: for any grid-aligned cap C,

$$0 \le \mathbb{E}\min(\Delta\lceil\tau^\pi/\Delta\rceil, C) - \mathbb{E}\min(\tau^\pi, C) \le \Delta.$$

This follows pointwise from $0 \le \Delta\lceil\tau/\Delta\rceil - \tau < \Delta$ at finite τ, with both capped quantities equal at $\tau = \infty$. Even with exact grid probabilities, the score measures time rounded up to the next block boundary, with the discrepancy bounded above.

Proposition 15 (Horizon-wise optima and policy consistency). *Let* Π *be a common policy class and* $V_j^*(x) = \sup_{\pi\in\Pi} F_j^\pi(x)$. *Define*

$$S_n^*(x) = \Delta \sum_{j=0}^{n} (1 - V_j^*(x)), \qquad C_n^*(x) = \inf_{\pi\in\Pi} S_n^\pi(x).$$

Then $S_n^* \le C_n^*$ *and, with* $A_\pi = \Delta \sum_{j=0}^{n} (V_j^* - F_j^\pi)$,

$$C_n^* - S_n^* = \inf_{\pi\in\Pi} A_\pi \ge 0. \tag{21}$$

Equality holds if a single policy achieves all the horizon optima. More generally, if some $\bar\pi$ *satisfies* $0 \le V_j^* - F_j^{\bar\pi} \le a_j$, *then* $0 \le C_n^* - S_n^* \le \Delta \sum_{j=0}^{n} a_j$.

Proof. For every fixed π, subtract the finite sums to obtain $S_n^\pi = S_n^* + A_\pi$. Each summand in A_π is nonnegative. Taking the infimum proves the identity and lower bound. Substituting a jointly optimal policy, or the approximate policy $\bar\pi$, gives the remaining claims. An infimum need not be attained for the identity to hold. □

A strict policy-consistency gap. Take $\Delta = 1$, $n = 2$, and an initial state with two irreversible choices. Choice A reaches the goal at time 1 with probability 0.6 and never reaches it otherwise. Choice B reaches the goal at time 2 with certainty. There are no later choices. Then $V_0^* = 0$, $V_1^* = 0.6$, and $V_2^* = 1$, so $S_2^* = 1.4$. Policy A has truncated expectation $0.6\cdot 1 + 0.4\cdot 3 = 1.8$; policy B has expectation 2. Randomizing between them gives a convex combination, so $C_2^* = 1.8 > S_2^*$. The values $j \mapsto V_j^*$ never decrease with the horizon and define a valid cumulative distribution on this grid. Yet no policy in this example has that hitting-time distribution. A critic can therefore be nondecreasing in the horizon without describing any single policy.

Expected truncated time is not deadline success. Even with a common policy interpretation, minimizing the score is not equivalent to maximizing probability of success by the final deadline. With deadline 5 and cap 6, one choice hits at time 1 with probability 0.49 and never hits otherwise; another hits at time 5 with certainty. Their scores are $0.49 + 0.51 \cdot 6 = 3.55$ and 5, respectively. The score prefers the less reliable choice because it assigns enough value to early success. Minimizing this cost therefore does not impose a lower bound on the probability of success by the deadline.

Corollary 16 (Clock sensitivity). *For* $n' \ge n$ *and any* $[0,1]$*-valued horizon predictions,*

$$0 \le \widehat{S}_{n'}(x) - \widehat{S}_n(x) \le \Delta(n' - n).$$

The difference of the score changes for two candidates has absolute value at most $\Delta(n' - n)$.

Proof. The added terms form a sum of $n' - n$ numbers in $[0, \Delta]$. Each candidate's increase lies in the same interval, so their difference lies between its negative and positive endpoints. □

Failing to update the remaining budget can therefore change candidate rankings. If reachability predictions increase with the horizon, querying beyond the available time can also overstate the probability of success by the actual deadline. Changing the horizon can additionally change the score IQR and its effective weight in the combined cost.

I.7 Finite-horizon Bellman analysis of the critic

On the stopped Markov block process, define

$$(\mathcal{B}f)(x) = \begin{cases} 1, & x \in \mathcal{G}_g, \\ \sup_{a\in\mathcal{A}(x)} \mathbb{E}[f(X') \mid x, a], & x \notin \mathcal{G}_g. \end{cases} \tag{22}$$

Here a is a Δ-step action block, and X' includes within-block success. Assume measurable policies and kernels for which the finite-horizon recursion is valid; finite state/action spaces suffice. Keep the same action class throughout the comparison. This operator describes optimal reachability using the allowed block actions. It need not equal optimal control with feedback at every primitive step.

Lemma 17 (Nonexpansiveness)**.** *For bounded f, g, $\|\mathcal{B}f - \mathcal{B}g\|_\infty \le \|f - g\|_\infty$. The corresponding fixed-policy operator has the same property.*

Proof. On the success set both outputs equal one. Elsewhere every conditional expectation differs by at most $\|f-g\|_\infty$. The inequality $|\sup_a u_a - \sup_a v_a| \le \sup_a |u_a - v_a|$ proves the claim. Taking the expectation under a fixed action distribution gives the policy case. □

This operator does not contract errors by a discount factor. The recursion converges because each horizon depends only on the preceding horizon and we fix the value at horizon zero.

Proof of Theorem 3. Set $e_j = \|\hat{V}_j - V_j^*\|_\infty$. Insert $\mathcal{B}\hat{V}_{j-1}$ and use Lemma 17 to obtain $e_j \le \xi_j + e_{j-1}$. Induction gives $e_j \le \varepsilon_0 + \sum_{i=1}^{j} \xi_i$. The triangle inequality bounds the survival-score difference by $\Delta \sum_{j=0}^{n} e_j$. Reordering the finite double sum yields

$$\sum_{j=0}^{n} \sum_{i=1}^{j} \xi_i = \sum_{i=1}^{n} (n - i + 1)\xi_i,$$

proving Equation (12). For the sweep claim, hold $V_0^{(k)} = V_0^*$ and update $V_j^{(k+1)} = \mathcal{B}V_{j-1}^{(k)}$. After the first sweep V_1 is exact, after the second V_2 is exact, and so on. An induction in j proves exactness through horizon n after n sweeps. □

If $\xi_j \le \xi$ uniformly, the score-error bound becomes $\Delta[(n+1)\varepsilon_0 + n(n+1)\xi/2]$. We can tighten it by replacing each e_j with $\min(1, \varepsilon_0 + \sum_{i \le j} \xi_i)$ when both values lie in $[0, 1]$. Probability error therefore grows at most linearly with the horizon under a uniform residual bound; summing those errors across horizons gives a quadratic bound.

Decomposing a residual of the actual bootstrap. Let $\mathcal{B}_M$ be the exact-model Bellman operator restricted to the sampled action blocks, and let $\widehat{\mathcal{B}}_M$ use predicted endpoints and the clipped pessimistic ensemble target. For a target function $\bar{V}_{j-1}$, the triangle inequality gives

$$\begin{aligned}\|\hat{V}_j - \mathcal{B}\hat{V}_{j-1}\|_\infty \le{}& \underbrace{\|\hat{V}_j - \widehat{\mathcal{B}}_M \bar{V}_{j-1}\|_\infty}_{\text{fitting and mixed-label error}} + \underbrace{\|\widehat{\mathcal{B}}_M \bar{V}_{j-1} - \mathcal{B}_M \bar{V}_{j-1}\|_\infty}_{\text{model and pessimism error}} \\ &+ \underbrace{\|\mathcal{B}_M \bar{V}_{j-1} - \mathcal{B}\bar{V}_{j-1}\|_\infty}_{\text{action-coverage error}} + \underbrace{\|\bar{V}_{j-1} - \hat{V}_{j-1}\|_\infty}_{\text{target lag}}. \end{aligned} \tag{23}$$

The last term follows from nonexpansiveness. Take suprema over a domain that contains every successor of each compared transition. If the initial domain is smaller, extend the error bound to every queried successor. A finite training loss alone does not bound these worst-case errors.

For example, in deterministic dynamics suppose the true and predicted block endpoints are within e_P, the target value is L_V-Lipschitz on the comparison domain, and the ensemble spread is at most $\bar{\sigma}$. Assume both operators enforce the same goal boundary and their endpoint descriptions include the within-block hit flag. The model-and-pessimism term is at most $L_V e_P + \kappa\bar{\sigma}$: each action value changes by at most this amount, clipping to $[0, 1]$ is nonexpansive, and taking a maximum preserves the bound. For a stochastic transition, a deterministic mean prediction generally does not preserve the expectation of a nonlinear value function. One must instead bound the error in the expected next-state value, for example using a total variation bound for $[0, 1]$-valued functions. Include that error in the same model residual term.

Finite action maximization. If, for a fixed query, a proposal hits an ϵ_A-optimal block with probability at least p_A, M independent draws miss that set with probability $(1 - p_A)^M$. On a hit, the restricted maximum is within ϵ_A of the full maximum for that query. The conditional version of Proposition 11 also applies. Sampling around logged actions does not by itself supply a positive lower bound on coverage of all optimal actions. A uniform Bellman residual must account for this restriction.

Boundary and absorbing-goal errors. The recursion requires $V_0 = \mathbf{1}_g$ and the value one on already-successful states at every horizon. If a target omits either boundary condition, include the resulting error in ε_0 or ξ_j. Hindsight at a logged future goal gives a successful behavior trajectory, but observing no hit by a deadline does not prove no other policy can hit. Similarly, an object displacement larger than any displacement observed in 50 steps does not prove physical impossibility in 50 steps. The mixed training procedure therefore approximates reachability, and Theorem 3 accounts for its error through the Bellman residual.

I.8 From critic calibration to candidate ranking

The graph theorem bounds the critic-score difference by ω without requiring accurate probabilities. When the critic does estimate reachability accurately, we can relate this difference to the true endpoint scores. Fix any true endpoint score $S(u)$ of interest: a specified policy's truncated expectation, or the score formed from the optimal reachability values at each horizon. Suppose on all compared sequences

$$|\widehat{S}(u) - S(u)| \le e_S, \qquad |\widehat{b}(u) - b(u)| \le e_b. \tag{24}$$

For the graph branch, $e_b \le e_\psi$ suffices. If critic error at true endpoints at horizon j is at most ε_j and the estimated critic is L_j-Lipschitz between the true and predicted endpoints, then an endpoint prediction bound e_z gives

$$e_S \le \Delta \sum_{j=0}^{n} (\varepsilon_j + L_j e_z). \tag{25}$$

This follows by adding and subtracting each critic value at its true endpoint and summing absolute differences. Checking critic accuracy only on encoded dataset frames does not control the additional error at predicted endpoints.

Proposition 18 (Candidate regret with fixed population scales)**.** *Fix positive scales and define $J(u) = b(u)/s_b + \beta S(u)/s_S$ and $\widehat{J}(u) = \widehat{b}(u)/s_b + \beta \widehat{S}(u)/s_S$. On a comparison set $\mathcal{U}$, suppose Equation* (24) *holds and $\widehat{J}(\hat{u}) \le \inf_{u\in\mathcal{U}} \widehat{J}(u) + \epsilon_{\mathrm{opt}}$, with $\hat{u} \in \mathcal{U}$. Then*

$$J(\hat{u}) \le \inf_{u\in\mathcal{U}} J(u) + \epsilon_{\mathrm{opt}} + 2\left(\frac{e_b}{s_b} + \beta\frac{e_S}{s_S}\right). \tag{26}$$

For a comparison sequence $u^\circ \in \mathcal{U}$ that tracks the graph target, use $\omega = [S(u^\circ) - S(\hat{u})]_+ + 2e_S$ in Theorem 1, provided its other hypotheses hold.

Proof. Let $e_J = e_b/s_b + \beta e_S/s_S$. Then $|J(u) - \widehat{J}(u)| \le e_J$ throughout $\mathcal{U}$, so

$$J(\hat{u}) \le \widehat{J}(\hat{u}) + e_J \le \inf_{\mathcal{U}} \widehat{J} + \epsilon_{\mathrm{opt}} + e_J \le \inf_{\mathcal{U}} J + \epsilon_{\mathrm{opt}} + 2e_J.$$

No minimizer need exist because the inequalities preserve infima. For the last claim, $\widehat{S}(u^\circ) - \widehat{S}(\hat{u}) \le S(u^\circ) - S(\hat{u}) + 2e_S$. Take positive parts and use $[a+b]_+ \le [a]_+ + b$ for $b \ge 0$. □

This result compares costs within a fixed set of action sequences using shared scales. It does not compare against an unrestricted optimal RL policy or imply greater success by the deadline. For our method, it shows that the actual IQR scale ratio multiplies critic error in the graph-progress bound. Even small probability errors can matter when the critic score has a very small normalization scale.

Proposition 19 (When the critic resolves ambiguous subgoals)**.** *Partition a comparison set $\mathcal{U}$ into acceptable and unacceptable plans, and let u° be one acceptable plan. Assume Equation* (24) *and shared positive scales. For every unacceptable u, suppose*

$$b(u) - b(u^\circ) \ge -a_b, \qquad S(u) - S(u^\circ) \ge m_S,$$

where $a_b \ge 0$ and $m_S > 0$. If the returned sequence belongs to $\mathcal{U}$, is within ϵ_{opt} of u° in predicted combined cost, and

$$\beta\frac{m_S - 2e_S}{s_S} > \frac{a_b + 2e_b}{s_b} + \epsilon_{\mathrm{opt}}, \tag{27}$$

then the returned plan is acceptable. If every acceptable plan tracks the graph target within $\rho_0 \le r_{\mathrm{TD}}$, $\rho_0 < \lambda$, the graph-descent conclusion holds with ρ_0 in place of Equation (9)*.*

Proof. For an unacceptable plan, subtract the predicted objective at u°. The shared medians cancel, and the error bounds give

$$\widehat{J}(u) - \widehat{J}(u^\circ) \geq -\frac{a_b + 2e_b}{s_b} + \beta\frac{m_S - 2e_S}{s_S} > \epsilon_{\text{opt}}.$$

This contradicts the stated comparison inequality if the returned plan is unacceptable. The final claim is Lemma 9. □

This condition shows how the critic can help select a useful plan. The base distance may favor an unacceptable plan by up to a_b. If the critic estimates the survival-score gap accurately enough and has enough weight, it reverses that ordering. For a common continuation policy, if an acceptable endpoint has grid hitting time at most $t_0 \leq n\Delta$ almost surely and an unacceptable endpoint cannot hit by $n\Delta$, their score gap is at least $C_n - t_0$. By contrast, if every candidate is unable to hit before truncation, all exact scores are C_n and the critic has no ranking margin. This gives a concrete reason for combining a short-horizon critic with graph direction. Applying these inequalities requires a true score gap and access to acceptable action sequences. Ensemble training alone does not guarantee either condition.

Supervised labels and population risk. Binary cross-entropy is a strictly proper scoring rule (Gneiting & Raftery, 2007). It identifies the conditional probability of the training labels, which need not equal optimal reachability. This distinction can be quantified.

Proposition 20 (Label bias and distribution shift)**.** *Let μ be a distribution of state–goal–horizon queries, let $q(w) = \mathbb{E}[Y \mid w]$, and let $\hat{V}(w) \in (0, 1)$ be any predictor. The label Y may be binary or a soft target in $[0, 1]$. Let R_{CE} be its population cross-entropy excess over q. For a query distribution ν with $d\nu/d\mu \leq C_\mu$ and a target reachability $F(w)$,*

$$\mathbb{E}_\nu|\hat{V} - F| \leq \sqrt{C_\mu R_{\text{CE}}/2} + \mathbb{E}_\nu|q - F|. \tag{28}$$

Proof. Conditional cross-entropy is linear in Y, so replacing Y by its conditional mean gives excess $\text{kl}(\text{Ber}(q) \,\|\, \text{Ber}(\hat{V}))$. For fixed $\hat{v} \in (0, 1)$, this divergence as a function of q has value and first derivative zero at $q = \hat{v}$, and second derivative $1/[q(1 - q)] \geq 4$ on $(0, 1)$. Twice integrating this curvature inequality, and taking limits at the endpoints, gives $\text{kl}(q\|\hat{v}) \geq 2(q - \hat{v})^2$. Therefore $\mathbb{E}_\mu(\hat{V} - q)^2 \leq R_{\text{CE}}/2$. Change measure, use Cauchy–Schwarz, and then the triangle inequality:

$$\mathbb{E}_\nu|\hat{V} - F| \leq \sqrt{\mathbb{E}_\nu(\hat{V} - q)^2} + \mathbb{E}_\nu|q - F| \leq \sqrt{C_\mu R_{\text{CE}}/2} + \mathbb{E}_\nu|q - F|.$$

□

Increasing network capacity or training time does not remove label bias. Hindsight behavior labels, selected cross-episode negatives, and pessimistic model targets can induce different conditional targets q. The result bounds the cross-entropy component for a fixed predictor and label distribution. The training procedure also uses a monotonicity penalty and moving targets, so it does not simply minimize that fixed risk. Moreover, a small average error does not give the uniform bound required by Proposition 18. CEM may select rare states where the critic makes large errors.

Proposition 21 (Finite-query calibration audit)**.** *Fix a trained critic and m query states, selected independently of a validation sample. For each query collect N independent trajectories under a specified common continuation policy, and form empirical first-hit probabilities $\bar{F}_j$ for $j = 0, \ldots, n$. With probability at least $1 - \delta$, simultaneously over all queries and horizons,*

$$|\bar{F}_j - F_j| \leq \sqrt{\frac{\log(2m(n + 1)/\delta)}{2N}}.$$

If the critic differs from each empirical value by at most ϵ_{fit}, its score error on these queries is at most C_n times the sum of ϵ_{fit} and the displayed radius.

Proof. For each fixed query and horizon, the N hit indicators are independent bounded Bernoulli variables. Apply Hoeffding's inequality (Hoeffding, 1963) and a union bound over $m(n + 1)$ averages. Different horizons of the same trajectory may be dependent; the union bound does not require independence between those averages. The triangle inequality bounds critic error, and summing the $n + 1$ errors with factor Δ proves the final claim. □

This proposition gives a validation procedure; we do not report such an experiment here. The bound applies only to the tested policy and queries. It does not establish accuracy for optimal reachability or for arbitrary future CEM queries. Applying it would require the independent validation trajectories described above.

I.9 Convergence of tabular expectile TD

Expectile regression (Newey & Powell, 1987) underlies implicit Q-learning (Kostrikov et al., 2022) and the HILP/GAS representation objective. We give the standard contraction proof for an unrestricted value table to explain what expectile TD guarantees and why that guarantee does not directly apply to neural TDR training.

For a bounded random variable Y and $\tau \in (0,1)$, let $e_\tau(Y) = \arg\min_v \mathbb{E}[\ell_\tau(Y - v)]$. Strict convexity of the loss gives uniqueness. Its first-order characterization is

$$\tau\mathbb{E}[(Y - e_\tau(Y))_+] = (1-\tau)\mathbb{E}[(e_\tau(Y) - Y)_+]. \tag{29}$$

The characterization also holds for degenerate Y.

Lemma 22 (Expectile monotonicity and stability)**.** *If $Y \le Z$ almost surely then $e_\tau(Y) \le e_\tau(Z)$. For a constant c, $e_\tau(Y + c) = e_\tau(Y) + c$. Consequently, if $|Y - Z| \le d$ almost surely, $|e_\tau(Y) - e_\tau(Z)| \le d$.*

Proof. Define $h_Y(v) = \tau\mathbb{E}[(Y-v)_+] - (1-\tau)\mathbb{E}[(v-Y)_+]$. It is continuous and strictly decreasing in v and crosses zero at $e_\tau(Y)$. When $Y \le Z$, both positive-part terms imply $h_Y(v) \le h_Z(v)$ for every v. Their zeros therefore have the same order. Adding c to both the random variable and the candidate value leaves every residual unchanged, proving the shift identity. Finally $Z - d \le Y \le Z + d$ and the first two claims give the stability bound. □

For the next proposition only, use a finite deterministic MDP with absorbing goal, one-step reward -1 off the goal, discount $\gamma \in (0,1)$, and fixed offline action probabilities $\mu(a \mid x)$ on nonempty supported action sets. On functions $U(g) = 0$, define

$$(\mathcal{T}_\tau U)(x) = \begin{cases} 0, & x = g, \\ e_\tau\{-1 + \gamma U(f(x,A)) : A \sim \mu(\cdot \mid x)\}, & x \ne g. \end{cases} \tag{30}$$

We fix the goal value at zero, including when a transition reaches the goal. The deterministic assumption matters: in a stochastic MDP, taking an upper expectile of sampled transitions can favor lucky random outcomes as well as better actions. It need not equal maximizing expected action values.

Proposition 23 (Tabular expectile convergence and residual bound)**.** *$\mathcal{T}_\tau$ is a γ-contraction in sup norm with a unique fixed point U_τ. Its iterates satisfy*

$$\|U^{(k)} - U_\tau\|_\infty \le \gamma^k \|U^{(0)} - U_\tau\|_\infty.$$

For any approximate value $\tilde{U}$ that is zero at the goal and satisfies $\|\tilde{U} - \mathcal{T}_\tau\tilde{U}\|_\infty \le \epsilon_{\mathrm{TD}}$,

$$\|\tilde{U} - U_\tau\|_\infty \le \epsilon_{\mathrm{TD}}/(1-\gamma). \tag{31}$$

Proof. Use the same sampled action A in both backups. Their sampled targets differ by at most $\gamma\|U - W\|_\infty$. Lemma 22 gives the contraction, with identical zero values at the goal. The finite-dimensional space of functions that are zero at the goal is complete in sup norm, so the contraction mapping theorem gives a unique fixed point and the iterate bound. For the residual claim, add and subtract $\mathcal{T}_\tau\tilde{U}$:

$$\|\tilde{U} - U_\tau\|_\infty \le \epsilon_{\mathrm{TD}} + \gamma\|\tilde{U} - U_\tau\|_\infty.$$

Rearrange using $1 - \gamma > 0$. □

Proposition 24 (Finite-expectile bias)**.** *Let U_* be the optimal discounted value restricted to the supported actions. Suppose some maximizing action at each nonterminal state has sampling probability at least $p_{\min} > 0$. If the span of the action targets $-1 + \gamma U_*(f(x,a))$ at each state is at most R_Q, then*

$$\|U_\tau - U_*\|_\infty \le \frac{(1-\tau)R_Q}{\tau p_{\min}(1-\gamma)}. \tag{32}$$

Also $U_\tau \le U_*$ pointwise.

Proof. Let M and m be the largest and smallest action targets at a state and let $v = e_\tau(Y) \in [m, M]$. A maximizing action has probability at least $p_{\min}$. By Equation (29),

$$\tau p_{\min}(M - v) \le \tau \mathbb{E}[(Y - v)_+] = (1 - \tau)\mathbb{E}[(v - Y)_+] \le (1 - \tau)(M - m).$$

Thus the expectile backup of U_* differs from its maximum backup by at most $(1 - \tau)R_Q/(\tau p_{\min})$. Since the maximum backup of U_* is U_*, apply Proposition 23's residual bound. For the ordering, $\mathcal{T}_\tau U \le \mathcal{T}_* U$ and both operators are monotone. Starting expectile iteration from U_* gives a nonincreasing sequence. Its limit is U_τ by contraction, proving $U_\tau \le U_*$. □

Why this is not a neural convergence theorem. The actual TDR restricts U to negative symmetric Euclidean distances, uses two networks and lagged targets, and optimizes a sampled nonconvex loss. Its updates are not the exact unrestricted iterations in Proposition 23. Even a perfectly optimized distance model may not represent U_τ. If finite directed distances satisfy $d_*(x, g) \ne d_*(g, x)$, any symmetric approximation d obeys

$$\max\{|d(x, g) - d_*(x, g)|,\ |d(g, x) - d_*(g, x)|\} \ge \tfrac{1}{2}|d_*(x, g) - d_*(g, x)|.$$

Subtract the two true distances and apply the triangle inequality to the errors to obtain this bound. HILP discusses this limitation of symmetric representations (Park et al., 2024), and quasimetric learning treats directionality directly (Wang et al., 2023). High expectile, twin networks, or longer optimization do not remove it.

Even in the ideal deterministic shortest-path setting, a d_*-step path has discounted cost $(1 - \gamma^{d_*})/(1 - \gamma)$, rather than d_*. The discounted cost saturates with path length. Empirical median calibration u_k sets useful working scales but does not make the representation accurate everywhere. We therefore analyze the finite graph and local execution errors without assuming that neural TD learns an exact temporal metric.

I.10 A SEPARATION FROM TERMINAL L2 AND THE ROLE OF GEOMETRY

Proposition 25 (An exact-model L2 trap at any fixed planning horizon)**.** *For every integer $K \ge 1$ and block duration $\Delta \ge 1$, there is a deterministic problem with an exact predictor, a connected graph, and equal-radius latent observations such that exact K-block terminal-L2 optimization keeps the agent at its start forever, even though the goal is reachable. The controller executes each full plan before replanning. Exact optimization of the graph-guided objective with its local switch and an exact reachability score succeeds in two calls, for every positive pair of normalization scales and $\beta \ge 0$.*

Proof. Use states $x_0, \dots, x_{2K} = g$. At each nonterminal state a block action either stays or advances one state; g is absorbing. A block lasts Δ primitive steps, with its state change at the last step. The budget is at least $2K\Delta$. Define injective raw latent observations on the unit circle by

$$z_g = (1, 0), \qquad z_{x_i} = (\cos\theta_i, \sin\theta_i), \quad \theta_i = \frac{\pi}{3} + \frac{\pi i}{3K}, \quad 0 \le i < 2K.$$

The start has squared L2 distance one from the goal. Every intermediate state has strictly larger distance, since $\pi/3 < \theta_i < \pi$ for $0 < i < 2K$ and $\|z_{x_i} - z_g\|^2 = 2 - 2\cos\theta_i$ increases on that interval. From x_0, the endpoints reachable within K blocks are precisely $x_0, \dots, x_K$; the goal is unavailable. The unique L2-minimizing endpoint is therefore x_0, achieved by staying throughout the call. Replanning repeats the same choice indefinitely.

Set $\psi(x_i) = i$, $r_{\mathrm{TD}} = 1$, and $\lambda = K$ in calibrated TDR units. Place singleton regular graph nodes at $x_0, \dots, x_{2K-1}$; the stated goal attachment rule connects g to x_{2K-1}. The radius graph has unit edges between successive states, so $D_g[x_i] = 2K - i$. At the start, either tied attachment x_0 or x_1 gives total path length $2K$, and its first λ-crossing target is x_K. An endpoint x_j, $0 \le j \le K$, has base cost $K - j$. Under the fastest continuation policy, its grid hitting time is $T_j = (2K - j)\Delta$. For every clock index $n \ge 0$, the exact survival score $\min(T_j, C_n)$ is nonincreasing in j. Thus the all-advance sequence ending at x_K strictly minimizes the base cost and weakly minimizes the critic cost. Shared centering and positive scaling preserve its strict combined-cost advantage for every $\beta \ge 0$.

At x_K, $d_\psi(x_K, g) = K = \lambda$, so the local L2 branch applies. A second all-advance sequence reaches g in K blocks and has both base cost and critic score zero. All other endpoints have strictly positive base cost and nonnegative critic score. Exact optimization therefore reaches the goal on this second call. □

Taking $K = 5$ and $\Delta = 5$ gives the paper's 25-step plan and execution cadence, with success in 50 steps. Appending zero coordinates embeds the construction in higher dimensions. We supply its radius and representation as fixed controller inputs; we do not claim that neural training recovers them. Both planners optimize their costs exactly in this example, so it isolates the effect of the objective. Finite-sample CEM may miss the best sequence, and the empirical tasks need not have this structure. The example therefore does not imply that every graph improves on L2.

Why a Gaussian shell is not a failure theorem. If z_1, z_2 are independent draws from $\mathcal{N}(\mu, \sigma^2 I_d)$, then

$$z_1 - z_2 \sim \mathcal{N}(0, 2\sigma^2 I_d), \qquad \tfrac{1}{2}(z_1 + z_2) - \mu \sim \mathcal{N}(0, \tfrac{1}{2}\sigma^2 I_d).$$

Therefore independent-pair midpoint radii have the same distribution as a single centered radius divided by $\sqrt{2}$, and pairwise distances have the radius distribution multiplied by $\sqrt{2}$. This follows directly from sums of independent Gaussian vectors. It motivates the interpolation diagnostic, but says nothing about which action sequences a predictor can execute. On a sphere of radius R, two points separated by angle $\theta \in [0, \pi]$ have chord distance $2R\sin(\theta/2)$, strictly increasing with the shorter arc length $R\theta$. Thus chord and arc distances can give identical rankings. A shell alone does not imply a defective planning cost.

Similarly, a narrow distribution of start–goal L2 distances does not prove that small candidate-dependent changes in that distance contain no useful signal. The latent-ranking measurements examine that question on the selected candidates. The proposition gives a concrete failure mechanism: the available actions require the agent to increase L2 distance before it can reach the goal.

I.11 What the guarantees establish and how they can fail

The combined theorem bounds the time to reach the goal under explicit assumptions. It does not prove convergence of a training algorithm. Each assumption identifies a way the controller can fail:

1. **A finite graph route is required.** If no eligible node connects to the goal, then $\Phi(z) = \infty$. Enlarging the goal radius adds a nearby attachment but does not ensure an executable connection.
2. **The selected target must be executable in one call.** The comparison sequence must end within η of the target. Median calibration does not guarantee this, and the first-crossing rule can select a target beyond the lookahead distance.
3. **Model-error bounds must cover both compared sequences.** Error under logged actions alone does not bound error on CEM's selected actions. In stochastic systems one must also bound the relevant expected or conditional execution error.
4. **Search and critic tradeoffs must preserve tracking.** The quantities $s_b\epsilon_{\mathrm{opt}}$ and $\beta(s_b/s_S)\omega$ both increase ρ. If the critic strongly favors plans that miss the target, the controller may fail to make graph progress.
5. **The local controller must stay in its region and reach the goal.** The limiting error bound must lie below the success tolerance, and sufficiently small latent error must imply the environment's goal condition.

One can check these conditions on the implemented controller, but the reported tables do not estimate all of them at every required state. The theory therefore explains how the method can succeed without certifying each evaluated episode numerically. Poor target tracking, a conflicting critic, or both could explain the Cube ablations; the aggregate results do not distinguish these causes. The full method need not outperform TDR, graph-only planning, or terminal L2 in every environment.

Our contribution is to combine the algorithm's restricted graph attachment, first-crossing targets, median/IQR normalization, full-plan execution, and repeated switching decision in the bounds Equations (9) to (11). We use the standard tail-sum identity, Bellman nonexpansiveness, expectile con-

traction, and concentration inequalities to support that analysis. The policy-consistency gap and counterexamples show why their assumptions matter for this controller.