LEARNING TO PLAN FROM RANDOM EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Random exploration reveals how an environment can be traversed before a goal is specified. Can this experience support long-range planning without policy-improvement training? Our random-walk analysis explains what temporal relations contain: short horizons reveal geodesic geometry in the diffusion limit, while longer horizons reveal connectivity between regions before mixing removes these distinctions. We learn these relations with a conditional energy-based model that estimates temporal log-density ratios through horizon-conditioned embeddings. The model is trained on observation pairs by noise-contrastive estimation, without action or reward labels. The planner queries these learned relations at different horizons as it moves toward the goal. At test time, a separate local dynamics model predicts candidate action outcomes, and the temporal model evaluates their progress toward the goal by selecting or aggregating estimated improvements across horizons. The agent executes one action and replans with both models fixed. Experiments demonstrate long-range maze planning from random exploration using states and images. Learned score fields, embedding probes, and planned routes exhibit properties of a multiscale cognitive map. We further demonstrate egocentric navigation from random exploration and manipulation planning from suboptimal data.

1 INTRODUCTION

In a classic latent-learning experiment, rats explored a complex maze for ten days with no food waiting at the goal (Tolman & Honzik, 1930). After food was introduced, rats that had wandered unrewarded took nearly as few wrong turns on subsequent trials as rats rewarded from the start. Exploration had taught them far more than their behavior had revealed. The cognitive-map hypothesis and studies of hippocampal place cells link navigation to internal spatial representations (Tolman, 1948; O'Keefe & Dostrovsky, 1971; Zhao et al., 2025b), while sequence-learning accounts explain how these representations can arise from temporal experience (Raju et al., 2024). We ask whether even *random* exploration is enough: can representations learned from trajectories collected without goal-directed or curiosity-driven action selection support long-range planning?

A random walk through an environment leaves temporal traces at multiple scales. Observations a few steps apart reveal local connections between states, like reading a map at street level. Observations many steps apart reveal connections between regions, like reading it at district level. We show that, under random-walk assumptions, short-horizon statistics recover geodesic geometry in the diffusion limit while longer horizons capture regional connectivity before mixing erases these distinctions. Classical results on heat kernels and diffusion processes (Varadhan, 1967; Norris, 1997; Coifman & Lafon, 2006) provide the basis for this analysis. The challenge is to learn these relations from sampled trajectories, without access to the full transition kernel, and let a planner query different scales as it moves toward a goal.

Existing methods obtain long-range guidance in different ways. Goal-conditioned reinforcement learning learns values and trains policies to act on them (Eysenbach et al., 2022; Park et al., 2023). Map-like representations can emerge through navigation objectives (Wijmans et al., 2023). Latent world models learn action-conditioned predictions from reward-free trajectories and evaluate multistep action sequences at test time (Zhou et al., 2025; Wang et al., 2026), but their planners depend on rollout accuracy and a latent geometry that reflects goal progress. Temporal contrastive representations can encode probability ratios that support planning via interpolation in representation space (Eysenbach et al., 2024). We separate long-range guidance from local dynamics prediction: a representation

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.

learned across horizons evaluates progress toward the goal, while a one-step dynamics model predicts the outcomes of candidate actions. Planning requires neither a trained goal-reaching policy nor multistep dynamics rollouts.

We model temporal relations with a conditional energy-based model (LeCun et al., 2006; Xie et al., 2016) whose source and target embeddings estimate temporal log-density ratios at each horizon. Noise-contrastive estimation (Gutmann & Hyvärinen, 2010) learns them from observation pairs, requiring no action or reward labels. At test time, the dynamics model predicts immediate action outcomes and the temporal representation scores their estimated improvement in goal likelihood. The temporal representation is reusable across goals, and the agent replans after each action with both models fixed.

Our contributions are threefold:

(1) We develop a conditional temporal model that learns multi-horizon representations from exploration trajectories and supports closed-loop goal reaching without policy-improvement training.
(2) We characterize the geodesic and connectivity information in the ideal random-walk temporal score, and unify greedy horizon selection and potential ascent through a temperature-controlled planning objective.
(3) We demonstrate long-range maze planning from random exploration using states and images. Score fields, embedding probes, and planned routes reveal properties of a multiscale cognitive map. Further evaluations cover egocentric navigation and manipulation with suboptimal data.

2 METHOD

Exploration trajectories reveal how an environment can be traversed before a goal is specified. We learn these temporal relations at multiple horizons, then query them to turn predictions of local action outcomes into goal-directed decisions.

2.1 PROBLEM FORMULATION

Given a dataset of trajectories in which each observation x_t is a state vector or an image, we aim to learn temporal relations that support goal-directed planning. Our focus is random exploration for navigation: locally feasible moves are sampled without reference to a task goal. The learning formulation also accepts suboptimal or expert trajectories, which can be useful in manipulation tasks. Data collection can therefore balance broad exploration with coverage of task-relevant transitions. In each case, we learn the temporal relations induced by the behavior represented in the dataset.

Let $\mathcal{T}$ be a finite set of positive prediction horizons. At horizon $\tau \in \mathcal{T}$, we sample a source–target pair $(x, y) = (x_t, x_{t+\tau})$ from the same episode, with distribution $p_{\mathrm{data}}(x, y|\tau)$. This distribution describes where exploration leads from x after τ steps, and varying τ describes this at different time scales. We model $p_{\mathrm{data}}(y|x, \tau)$ so that, once a goal is given, the planner can ask how likely each candidate next observation is to lead to it within τ steps. Learning this conditional uses only observation pairs, without action or reward labels. Action-labeled one-step transitions (x_t, a_t, x_{t+1}) are used separately to learn local dynamics.

2.2 MODELS

A multi-horizon representation. The temporal relation between two observations depends on how long the agent explores. We describe it through the alignment of a source embedding $h_\theta(x, \tau)$ and a target embedding $g_\theta(y, \tau)$. The family $\{h_\theta(\cdot, \tau), g_\theta(\cdot, \tau)\}_{\tau \in \mathcal{T}}$ represents how these relations change with the horizon. Two observations may have little association over a short horizon but become associated once exploration has had time to traverse a connecting route.

The planner can query this family at different horizons to evaluate the same candidate move at several temporal scales. In spatial environments, these relations can exhibit properties of a *multiscale cognitive map* (Tolman, 1948; Zhao et al., 2025b). Section 3 connects them to geodesic geometry and connectivity between regions under random-walk assumptions.

Conditional temporal model. We model the target observation y given the source x and horizon τ with a conditional energy-based model (EBM). Let $p_0(y)$ be a reference distribution whose support

covers the target observations. The model is

$$p_\theta(y|x,\tau) = \frac{p_0(y)}{Z_\theta(x,\tau)} \exp(\beta_0(\tau)\langle h_\theta(x,\tau), g_\theta(y,\tau)\rangle)\,, \tag{1}$$

where $Z_\theta(x,\tau) = \mathbb{E}_{p_0(y)}[\exp(\beta_0(\tau)\langle h_\theta(x,\tau), g_\theta(y,\tau)\rangle)]$ normalizes the conditional distribution. Both embeddings have unit norm, $\|h_\theta(x,\tau)\|_2 = \|g_\theta(y,\tau)\|_2 = 1$. One variant used in the Habitat experiments leaves g_θ unnormalized, so that its magnitude carries a per-target level (the free goal head of Tab. 11); every other experiment uses unit-norm embeddings on both sides. For a given source and horizon, targets with larger alignment receive greater weight relative to p_0. The learned coefficient $\beta_0(\tau)$ controls the strength of this dependence across horizons.

Parameterization. $h_\theta(x,\tau)$ and $g_\theta(y,\tau)$ are encoders that take an observation and a horizon as input. They receive a learned embedding of τ, so one network covers all horizons and an observation can have a different embedding at each. The architecture depends on whether inputs are states or images. We write θ for all encoder, horizon-embedding, and coefficient parameters.

Local dynamic model. The temporal model describes where exploration can lead. To evaluate an action, we also need a prediction of its immediate consequence. A separately parameterized local model predicts

$$\widehat{x}^a_{t+1} = F_\phi(x_t, a). \tag{2}$$

The hat distinguishes this prediction from the observation obtained after execution. The temporal representation then evaluates the predicted outcome through its relations to the goal.

2.3 Learning

Noise-contrastive estimation. At a given horizon, we distinguish observed temporal pairs from pairs formed by sampling the target observation independently from p_0. These provide positive and negative examples, respectively. The EBM in Eq. (1) expresses temporal association through the log-density ratio

$$\begin{aligned}\log \frac{p_\theta(y|x,\tau)}{p_0(y)} &= \beta_0(\tau)\langle h_\theta(x,\tau), g_\theta(y,\tau)\rangle - \log Z_\theta(x,\tau) \\ &\approx \beta_0(\tau)\langle h_\theta(x,\tau), g_\theta(y,\tau)\rangle + \beta_1(\tau) =: G_\theta(x,y,\tau).\end{aligned} \tag{3}$$

When p_0 is the target marginal, the ratio compares how likely y is to follow x at horizon τ with how often y occurs in the data. The learned offset $\beta_1(\tau)$ approximates $-\log Z_\theta(x,\tau)$, allowing us to fit G_θ by noise-contrastive estimation (NCE) without evaluating the partition function (Gutmann & Hyvärinen, 2010; Ma & Collins, 2018). Because the offset is shared across sources, source-wise normalization is approximate; Sec. B.1 details this distinction.

With N negatives per positive and the same source marginal for both classes, the Bayes probability of a positive label is

$$\frac{p_{\text{data}}(y|x,\tau)}{p_{\text{data}}(y|x,\tau) + N p_0(y)} = \sigma\left(\log \frac{p_{\text{data}}(y|x,\tau)}{p_0(y)} - \log N\right), \tag{4}$$

where σ is the logistic sigmoid. Fitting $\sigma(G_\theta - \log N)$ therefore estimates the temporal log-density ratio. The reference enters through sampled negatives, so its density need not be evaluated.

We train across horizons by sampling τ from a chosen distribution $p(\tau)$ over $\mathcal{T}$ and minimizing

$$\begin{aligned}\mathcal{L}_{\text{NCE}}(\theta) = \mathbb{E}_{p(\tau)}\Big[&- \mathbb{E}_{p_{\text{data}}(x,y|\tau)}[\log \sigma(G_\theta(x,y,\tau) - \log N)] \\ &- N\, \mathbb{E}_{p_{\text{data}}(x|\tau)p_0(y)}[\log(1 - \sigma(G_\theta(x,y,\tau) - \log N))]\Big].\end{aligned} \tag{5}$$

The two terms distinguish temporal pairs from independent pairs, using the same source marginal $p_{\text{data}}(x|\tau)$. All temporal-model parameters are learned jointly across horizons. Our objective shares a similar loss form with SigLIP (Zhai et al., 2023).

At the unrestricted population optimum, $G^*(x,y,\tau) = \log[p_{\text{data}}(y|x,\tau)/p_0(y)]$ where both densities are positive. The constrained embedding model approximates this score, so e^{G_θ} estimates the conditional target density relative to the reference.

Learning the local dynamics. We train the local predictor on action-labeled transitions at horizon $\tau = 1$. Let $p_{\text{step}}(x_t, a_t, x_{t+1})$ denote their empirical distribution. For state observations, we minimize

$$\mathcal{L}_{\text{dyn}}(\phi) = \mathbb{E}_{p_{\text{step}}(x_t,a_t,x_{t+1})}\left[\|F_\phi(x_t, a_t) - x_{t+1}\|_2^2\right]. \tag{6}$$

The framework has a *hierarchical* organization: local dynamics predicts one-step action outcomes, which the multi-horizon representation evaluates against the goal across horizons. Both are learned before planning and remain fixed at test time. Implementation details are given in Sec. D.

2.4 Planning by Probability Improvement

Given a goal observation y_g, the planner favors actions whose predicted outcomes make the goal more likely. At the current observation x_t, the dynamics model predicts $\hat{x}^a_{t+1} = F_\phi(x_t, a)$ for each $a \in \mathcal{A}(x_t)$. We define progress at horizon τ as

$$\Delta_\tau(a) := e^{G_\theta(\hat{x}^a_{t+1}, y_g, \tau)} - e^{G_\theta(x_t, y_g, \tau)}. \tag{7}$$

By Eq. (3), this quantity estimates the increase in goal probability relative to $p_0(y_g)$, $\Delta_\tau(a) \approx [p_\theta(y_g|\hat{x}^a_{t+1}, \tau) - p_\theta(y_g|x_t, \tau)]/p_0(y_g)$. At a fixed horizon, the current-state term is shared by all actions, so ranking actions by $\Delta_\tau(a)$ is equivalent to ranking their predicted outcomes by $e^{G_\theta(\hat{x}^a_{t+1}, y_g, \tau)}$.

Combining horizons. The same move can make different progress at different horizons. We combine these improvements with a temperature $\beta > 0$, distinct from $\beta_0(\tau)$ and $\beta_1(\tau)$:

$$a^*_t \in \arg\max_{a \in \mathcal{A}(x_t)} \beta \log\left[\frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}} \exp\left(\frac{\Delta_\tau(a)}{\beta}\right)\right]. \tag{8}$$

As $\beta \to 0^+$, the objective becomes $\max_{\tau\in\mathcal{T}} \Delta_\tau(a)$. The planner greedily picks the action and horizon with the largest improvement, so each decision comes with an explicit horizon selection.

As $\beta \to \infty$, the objective becomes $|\mathcal{T}|^{-1}\sum_{\tau\in\mathcal{T}} \Delta_\tau(a)$. Unlike the greedy rule, which uses only the best horizon, averaging lets gains and losses at different horizons offset each other. The current-state term cancels, so candidates are ranked by $\sum_{\tau\in\mathcal{T}} e^{G_\theta(\hat{x}^a_{t+1}, y_g, \tau)}$, a single potential over states for a fixed goal. With exact predictions and positive progress at every step, this potential increases along the trajectory, so no state is revisited.

We refer to these two limits as *greedy planning* (GP) and *potential-ascent planning* (PAP), respectively, throughout the paper.

Closed-loop execution. At each step, it predicts the outcome of every candidate action, scores these outcomes by Eq. (8), and executes a^*_t. The agent then observes the actual x_{t+1} and repeats, until it reaches the goal or exhausts its budget. Because each prediction starts from an actual observation, errors do not compound over a multi-step rollout, while G_θ provides long-range guidance. Both θ and ϕ remain fixed throughout execution.

Section C summarizes learning θ, learning ϕ, and closed-loop planning.

3 Theoretical Understanding

We study what temporal relations learned from random exploration reveal about an environment. The results concern the calibrated score $G^*(x, y, \tau) = \log[p_{\text{data}}(y|x, \tau)/p_0(y)]$, the ideal log-density ratio our model approximates, and characterize the environmental structure it contains. Section B.1 relates this score to the learned model.

From random walks to the heat equation. Consider a symmetric local random walk over the free space, where a move into a wall leaves the agent in place. In free space, its steps are independent with zero mean and covariance $2\alpha I_n$. The constant $\alpha > 0$ is set by the one-step transition kernel and does not depend on position or horizon. After τ steps, the root-mean-square displacement is $\sqrt{2n\alpha\tau}$, so doubling this diffusion scale takes four times as many steps. For regular domains and lattice approximations satisfying a reflected-Brownian invariance principle (Burdzy & Chen, 2008), we study the limit under diffusive rescaling, with time per step proportional to squared step length. Writing τ also for this rescaled time, with diffusion coefficient α, its transition density satisfies $\partial_\tau p_{\text{data}} = \alpha\, \Delta_y p_{\text{data}}$.

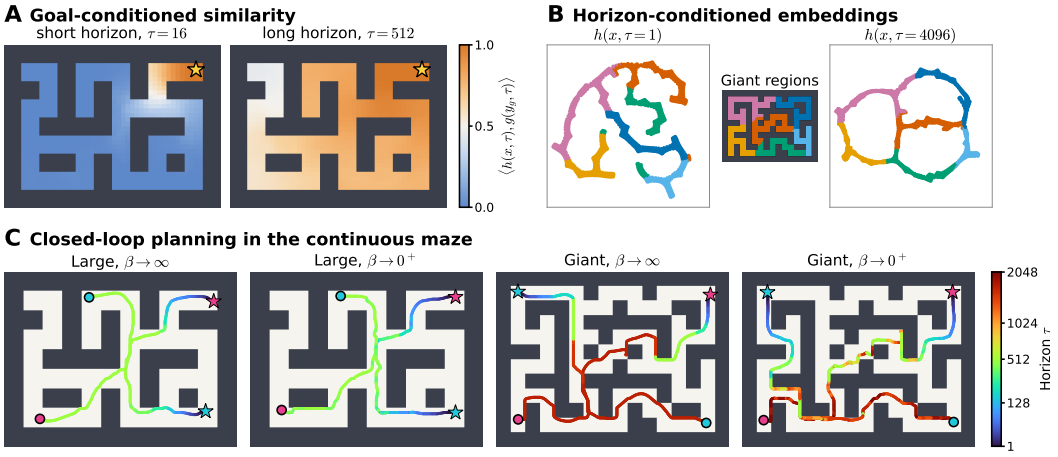

Figure 1: Emergent cognitive maps from random exploration. **(A)** Score $G_\theta(x, y_g, \tau)$ over positions x in Large, with goal y_g fixed (star). Short horizons ($\tau = 16$) peak near the goal, while long horizons ($\tau = 512$) spread through connecting passages. **(B)** t-SNE of $h_\theta(x, \tau)$ in Giant at $\tau = 1$ and 4096. At $\tau = 1$, embeddings preserve local geodesic geometry, while at $\tau = 4096$, they reflect connectivity between distant regions. 3D views appear in Fig. 7. **(C)** GP and PAP routes through Large and Giant. Colour denotes the selected horizon for GP and the dominant score-contributing horizon for PAP. Matching markers pair starts ($\circ$) and goals ($\star$).

Proposition 1 (Short horizons encode geodesic geometry)**.** *Consider the reflected Brownian motion above on a smooth, bounded, connected domain, with uniform p_0. Let $d_{\mathrm{geo}}(x, y)$ be the geodesic distance, the infimum of feasible path lengths between x and y. For fixed interior x, y,*

$$\lim_{\tau\downarrow 0} -4\alpha\tau\, G^*(x, y, \tau) = d_{\mathrm{geo}}(x, y)^2. \tag{9}$$

Proof sketch. Varadhan's formula gives $-4\alpha\tau \log p_{\mathrm{data}}(y|x, \tau) \to d_{\mathrm{geo}}(x, y)^2$ (Varadhan, 1967; Norris, 1997), and $4\alpha\tau \log p_0(y) \to 0$. □

This limit concerns rescaled diffusion time, rather than integer horizons on a fixed lattice (Sec. B.2). For the limiting diffusion, targets with shorter geodesic distances have larger scores at sufficiently short horizons, whatever the Euclidean distances. A nearby point behind a wall may require a long detour. At short horizons it then scores below a point that is farther in Euclidean distance but reachable along a shorter open route.

We now return to the discrete walk on a finite connected state space, with transition matrix P and stationary distribution π, which is uniform because P is symmetric.

Proposition 2 (Long horizons encode multiscale connectivity)**.** *Suppose P is irreducible, aperiodic, and reversible, with eigenvalues $1 = \lambda_1 \geq \lambda_2 \geq \cdots$ and eigenfunctions ψ_j orthonormal under π. Set $p_0 = \pi$. For every integer $\tau \geq 1$ and every pair with $P^\tau(x, y) > 0$,*

$$G^*(x, y, \tau) = \log\left[1 + \sum_{j\geq 2} \lambda_j^\tau\, \psi_j(x)\psi_j(y)\right]. \tag{10}$$

Proof sketch. Divide $P^\tau(x, y) = \pi(y) \sum_j \lambda_j^\tau \psi_j(x)\psi_j(y)$ by $\pi(y)$, separate $\psi_1 = 1$, and take the logarithm (Aldous & Fill, 2002; Coifman & Lafon, 2006). □

Terms with smaller $|\lambda_j|$ decay faster. Modes near $+1$ preserve differences between regions that exchange probability slowly. Picture two rooms joined by a narrow doorway. When movement between the rooms is slow relative to mixing within each room, a walk forgets which corner it started in long before it forgets which room it started in. Over these horizons, G^* distinguishes the rooms more strongly than positions within each room (Sec. B.3).

The useful scale changes during planning. Choosing a horizon is like choosing the zoom level of a map. Zoomed in, the map shows the next turn but may leave the connecting passage out of view. Zoomed out, the passage appears, but the turns leading to it blur together. A navigator zooms out when the destination is far and zooms in as it approaches, and the horizon plays the same role. Progress compares the true goal probability from each candidate outcome with that from the current state. At a horizon too short for the walk to reach the goal, these probabilities are all nearly zero, so every candidate makes almost no progress. As the horizon goes to infinity, they all approach the stationary value $\pi(y_g)$, and progress again vanishes. Only intermediate horizons separate good moves from bad ones. In free space, the best horizon for a small step is roughly the one at which the walk's typical displacement reaches the goal, so it grows with the squared goal distance (Sec. B.2). In a maze, it also depends on the time needed to cross connecting passages. Empirically, the greedy planner selects shorter horizons when approaching the goal (Fig. 1C).

4 Experiments

We evaluate goal reaching, horizon selection, data efficiency, first-person observations, and manipulation. Section 4.1 specifies the data and evaluation conditions and Sec. 4.2 reports the results.

4.1 Experimental Setup

Datasets and Environments. We study three settings. *(1) Maze navigation:* OGBench PointMaze Large and Giant (Park et al., 2025), with position or 64×64 overhead images. Our main models use 4M lattice transitions: eight neighbouring moves or stay sampled uniformly on a 1-unit grid, with blocked moves replaced by stay and five simulator steps per transition. Reproduced baselines use 1M transitions from uniformly sampled actions (20 episodes of 50k steps). Table 6 compares random-action and lattice training data; our local dynamics uses 1M transitions on Large and 10M on Giant. NAVIGATE uses OGBench expert data. *(2) Egocentric navigation:* a first-person Large maze and Habitat's Van Gogh Room and Apartment 1 (Savva et al., 2019). Maze inputs are images with optional position and/or heading. Habitat models use one 8M-step walk per scene; lattice planning compares position alone with images and position, and continuous planning adds heading. *(3) Manipulation:* OGBench cube-single with expert-only and mixed data. The expert-only setting follows LeWM's data and evaluation protocol (Maes et al., 2026) and uses its reported baseline results. The mixed setting combines expert trajectories with noisy oracle trajectories, generated by adding temporally correlated noise to the oracle's waypoint plan and replacing each action with a random one with probability 0.1.

Evaluation. We report success rate (SR) and success weighted by path length (SPL) (Anderson et al., 2018):

$$\mathrm{SR} = \frac{1}{M}\sum_{i=1}^{M} S_i, \qquad \mathrm{SPL} = \frac{1}{M}\sum_{i=1}^{M} S_i \frac{\ell_i}{\max(p_i, \ell_i)}, \tag{11}$$

Here M counts episodes, S_i indicates success, p_i is executed path length, and ℓ_i is shortest-path length to the goal region. Setting (1) uses five official tasks (20 episodes each, 1k-simulator-step budget) and 20 fixed start–goal pairs per maze (geodesic separation ≥ 20 cells, 5k-simulator-step budget). Success requires reaching within one unit of the goal. Candidate outcomes come from learned local dynamics, without simulator queries before acting (Sec. D.3). Setting (2) reuses the maze tasks. Habitat evaluates its two most distant reachable corner pairs in both directions: 25 episodes per task, random initial headings, and a 5k-lattice-action budget. Success requires reaching within one lattice move of the goal, continuous evaluation follows Sec. D.5. In setting (3), the goal is the observation 25 steps after the initial state in the same dataset trajectory. Each episode allows 50 steps and succeeds if the cube center comes within 0.04 m of its goal position at any step. We report SR only. Mixed-data comparisons with LeWM use 1,000 episodes from the held-out split. Further evaluation details and reporting conventions appear in Sec. D.

Comparisons. *World models* (LeWM, DINO-WM) plan through learned dynamics using their own planners. *Offline goal-conditioned RL* (GCIQL, HIQL, HILP, QRL) executes policies learned from offline data. Random actions and a shortest-path oracle provide reference controls. Baseline configurations appear in Sec. D.5.

Table 1: PointMaze goal reaching from states (A) and images (B) on official tasks and random start–goal pairs. The GCIQL, QRL, and HIQL NAVIGATE results are quoted from Park et al. (2025).

A State

	Official tasks						Random pairs			
	NAVIGATE		Random Data				Random Data			
	Large	Giant	Large		Giant		Large		Giant	
Method	SR↑	SR↑	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
Random policy (floor)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shortest-path oracle	1.00	1.00	1.00	0.94	1.00	0.98	1.00	0.95	1.00	0.95
GCIQL (Kostrikov et al., 2022)	0.34	0.00	0.00	0.00	0.00	0.00	0.15	0.15	0.07	0.06
QRL (Wang et al., 2023)	0.86	0.68	0.33	0.28	0.02	0.01	0.40	0.34	0.30	0.25
HIQL (Park et al., 2023)	0.58	0.46	0.00	0.00	0.00	0.00	0.12	0.11	0.10	0.09
HILP (Park et al., 2024)	—	—	0.49	0.47	0.00	0.00	0.47	0.45	0.18	0.18
LeWM (Maes et al., 2026)	0.08	0.00	0.10	0.03	0.00	0.00	0.45	0.14	0.10	0.04
PAP ($\beta \to \infty$)	—	—	0.75	0.66	**0.79**	**0.64**	0.92	0.83	**0.85**	**0.77**
GP ($\beta \to 0^+$)	—	—	**0.92**	**0.83**	0.76	0.58	**0.97**	**0.88**	**0.85**	0.75

B Vision

	Official tasks		Random pairs			
	Large	Giant	Large		Giant	
Method	SR↑	SR↑	SR↑	SPL↑	SR↑	SPL↑
LeWM (Maes et al., 2026)	0.00	0.00	0.00	0.00	0.00	0.00
DINO-WM (Zhou et al., 2025)	0.03	0.00	0.35	0.23	0.00	0.00
PAP ($\beta \to \infty$)	0.79	**0.53**	**0.95**	0.83	0.70	0.63
GP ($\beta \to 0^+$)	**0.91**	0.39	**0.95**	**0.85**	**0.78**	**0.67**

4.2 RESULTS

We first show that random exploration supports long-range planning from both states and images, and that the learned representation acquires multiscale spatial structure consistent with the random-walk analysis (Tab. 1, Fig. 1). We then study how the available horizons and exploration coverage shape planning performance (Fig. 2, Tab. 2). Finally, we test planning from egocentric observations and manipulation with suboptimal data (Fig. 3, Tab. 3).

Planning from random exploration. Temporal relations learned from random exploration support long-range goal reaching from both states and images (Tab. 1). Averaged over three training seeds, GP and PAP reach 0.97 and 0.92 of the random start–goal pairs on Large and 0.85 of them on Giant from states. From images, GP and PAP reach 0.95 of the random pairs on Large with SPL 0.85 and 0.83, and 0.78 and 0.70 of them on Giant. Most reported baselines achieve little or no success on the official tasks in the random-data setting. Unlike goal-conditioned RL, our model learns temporal relations under the exploration behavior without policy-improvement training. These relations provide long-range guidance for local action selection under both planning objectives.

Emergent cognitive maps. The learned temporal representation exhibits properties of a multiscale cognitive map (Tolman, 1948) in its scores, embeddings, and planned routes. For a fixed goal, short-horizon embedding similarities $\langle h_\theta(x,\tau), g_\theta(y_g,\tau)\rangle$ emphasize nearby locations, while longer horizons reveal connections through passages (Fig. 1A). The embeddings $h_\theta(x,\tau)$ represent the same environment at different spatial scales (Fig. 1B). Distance probes quantify this change: shorter horizons better preserve local geodesic geometry, while longer horizons better reflect connectivity between distant states (Tab. 4). These patterns are consistent with the random-walk analysis in Sec. 3 and emerge without distance or route supervision. In planning, GP and PAP use this structure to turn random exploration into efficient, goal-directed routes through the maze (Fig. 1C).

Horizon selection and aggregation. The larger maze benefits from longer planning horizons (Fig. 2): Large reaches about 0.8 success at $\tau_{\max} = 64$, whereas Giant improves mainly as the range grows from 256 to about 1,024. Further extension does not consistently improve performance. GP selects the largest progress at any horizon, while PAP aggregates progress across horizons. The two achieve comparable success on random maze pairs (Tab. 1), while PAP achieves higher success in the cube-single comparison (0.92 versus 0.84; Tab. 3). Along the routes in Fig. 1C, GP's selected horizon and PAP's dominant score-contributing horizon generally shorten near the goal, although PAP continues

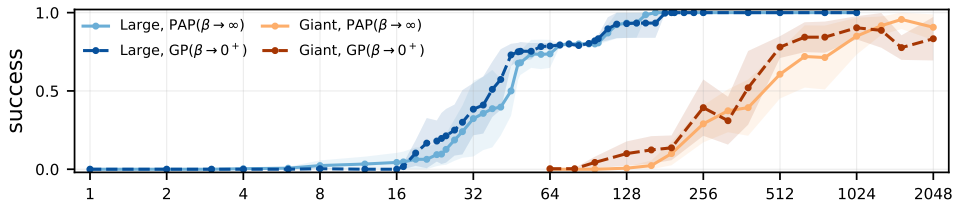

Figure 2: Official-task success versus maximum planning horizon. Color distinguishes Large and Giant and line style distinguishes GP and PAP. Lines show three-seed means and bands one standard deviation. Planning uses every integer horizon from 1 to $\tau_{\max}$.

Table 2: Goal reaching vs. walk length and exploration coverage c. Full coverage ($c = 1$) repeats the 4M reference.

	GP		PAP	
Walk (steps)	Large	Giant	Large	Giant
0.1M	0.45 $\pm$0.19	0.01	0.26 $\pm$0.18	0.00
0.4M	0.74 $\pm$0.06	0.15 $\pm$0.12	0.68 $\pm$0.16	0.13 $\pm$0.09
1M	0.91 $\pm$0.09	0.49 $\pm$0.09	0.68 $\pm$0.18	0.50 $\pm$0.21
4M	0.92 $\pm$0.09	0.76 $\pm$0.08	0.75 $\pm$0.13	0.79 $\pm$0.14
$c = 0.25$	0.09 $\pm$0.09	0.00 $\pm$0.00	0.27 $\pm$0.09	0.19 $\pm$0.16
$c = 0.50$	0.24 $\pm$0.17	0.03	0.26 $\pm$0.10	0.02
$c = 0.75$	0.56 $\pm$0.14	0.36	0.59 $\pm$0.22	0.20
$c = 1$	0.92 $\pm$0.09	0.76 $\pm$0.08	0.75 $\pm$0.13	0.79 $\pm$0.14

Table 3: Cube-single: mixed data, held-out episodes (top); expert data, LeWM's protocol and baselines (bottom).

Expert data ratio	PAP	LeWM
0%	0.908 $\pm$0.009	0.633 $\pm$0.013
25%	**0.933** $\pm$0.008	0.626 $\pm$0.009
50%	0.926 $\pm$0.006	0.623 $\pm$0.015
75%	0.905 $\pm$0.010	0.625 $\pm$0.010
100%	0.876 $\pm$0.010	**0.710** $\pm$0.012
PLDM (Sobal et al., 2025)	0.65	
LeWM (Maes et al., 2026)	0.74	
DINO-WM (Zhou et al., 2025)	0.86	
GP/PAP	0.84/**0.92**	

to aggregate all horizons. This change occurs without a prescribed schedule and is consistent with the spatial-scale analysis in Sec. 3.

Exploration coverage. Every walk in the length rows visits every free cell (a 0.1M-step walk already passes each Large cell about 136 times and each Giant cell about 73 times), so these rows vary the number of transitions per cell rather than coverage. The coverage fraction c specifies the part of the maze available to a 4M-step walk. Spreads denote standard deviations. The 4M and $c = 1$ rows are the models of Tab. 1. Every row plans with all horizons from 1 to 8192 through the learned local dynamics. Both data volume and state coverage affect planning performance. On Large, increasing the explored fraction from $c = 0.25$ to $c = 0.75$ raises success from 0.09 to 0.56 for GP and from 0.27 to 0.59 for PAP, and full coverage reaches 0.92 and 0.75 (Tab. 2). With unrestricted exploration, GP on Large rises from 0.45 after 0.1M steps to 0.91 after 1M steps. Giant needs longer walks: GP reaches 0.01 after 0.1M steps, 0.49 after 1M and 0.76 after 4M. Planning therefore improves with both walk length and coverage, and the larger maze requires more exploration.

Egocentric observations. Planning extends to first-person observations. In Habitat, we train on one 8M-step walk per scene, collected with 30° turns and 0.5 m forward or backward steps on a 0.25 m lattice (Sec. D.5). On Apartment 1's four corner tasks, images and position achieve 0.71 ± 0.12 success at $\tau_{\max} = 8{,}192$, compared with 0.38 for a random policy. The position-only planner repeatedly turns in place and reaches no goals (Tab. 9). Under the continuous protocol of Tab. 1 (36 headings, walking 0.75 m to the scored candidate before re-planning, no constraint on consecutive headings), the image-and-position model reaches a far pair of the apartment from all three start headings with horizons capped at 512 (SR 1.00, SPL 0.77; position alone: SR 1.00, SPL 0.79). The full horizon range strands the image model, since the longest horizons dominate at headings unseen in training, as on Giant beyond $\tau_{\max} \approx 1{,}000$ (Fig. 2); Sec. D.5 gives the per-model details. Along a successful route, the selected horizon shortens as the goal comes into view (Fig. 3). In the egocentric maze, position helps more than heading (Sec. D.5).

Manipulation with suboptimal data. Our model supports effective manipulation planning from suboptimal trajectories. On cube-single under LeWM's evaluation protocol, PAP reaches success 0.92 while GP reaches 0.84. We also vary the expert fraction at a fixed training budget and compare PAP with LeWM on matched data (Tab. 3). Our model achieves higher success at every mixture. Noisy-oracle data alone yields success 0.908, exceeding 0.876 with expert-only data, and the highest reported success is 0.933 at 25% expert data. Our temporal model learns how states are connected from the observed trajectories. Noise may expose a wider range of these connections, making suboptimal trajectories useful for planning even when they are less efficient at reaching goals.

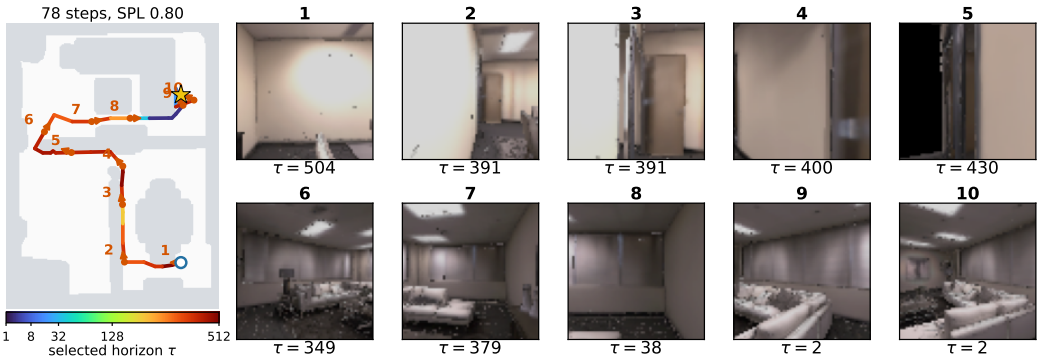

Figure 3: An egocentric plan in Habitat Apartment 1 by the model that plans from the image and the position (horizons capped at 512, no constraint on consecutive headings). Left: the route, coloured by the selected horizon, from the start (○) to the goal (⋆). Right: the agent's view at ten decisions and the horizon selected there.

5 Related Work

World models and temporal representations. Latent world models support goal reaching through action-sequence optimization. DINO-WM predicts dynamics in pretrained visual features (Zhou et al., 2025), while Temporal Straightening jointly learns representations and dynamics with a curvature regularizer (Wang et al., 2026). Temporal representation learning provides another source of structure. Contrastive predictive coding discriminates future observations (van den Oord et al., 2018), and Contrastive RL connects temporal associations to goal-conditioned values (Eysenbach et al., 2022). Successor and forward–backward representations relate future occupancy to rewards (Dayan, 1993; Touati & Ollivier, 2021), while contrastive representations support planning through interpolation and temporal reasoning (Eysenbach et al., 2024; Ziarko et al., 2025). Our model separates long-range guidance learned from observation pairs from local action-outcome prediction.

Random-walk geometry and planning. Diffusion maps connect transition statistics to geometry (Coifman & Lafon, 2006). Zhao et al. (2025b) model place-cell populations through position embeddings fitted to predefined random-walk kernels. We learn conditional temporal log-density ratios from sampled observation pairs through NCE, using horizon-conditioned state or image encoders. These scores guide action selection through horizon-specific progress in GP or a goal-dependent potential in PAP. Quasimetric learning targets directed optimal goal distances (Wang et al., 2023), while replay-buffer search composes local connections (Eysenbach et al., 2019). Our temporal scores describe exploration behavior, with geometric and connectivity interpretations under the random-walk assumptions in Sec. 3.

6 Limitations and Future Work

Random exploration can teach representations that support goal-directed behavior without policy-improvement training. Our current models learn within individual environments. Planning in unseen environments raises the question of what temporal structure transfers and what must be learned from new experience. Behavioral timescale synaptic plasticity supports rapid place-field formation (Bittner et al., 2017) and may inspire adaptation from limited experience. Egocentric planning presents a related challenge: a single view may not determine the state. Observation histories could help recover spatial context and reduce reliance on explicit position inputs. Scaling training to larger datasets across diverse environments would let us study whether shared temporal representations support transfer and rapid adaptation.

The connection to animal cognition also remains incomplete. Although the embeddings are learned, the learning objective, horizon conditioning, and planning rules are designed explicitly. Our framework does not explain how animals acquire these mechanisms. A broader account would need to explain how evolution shapes general learning and control mechanisms through which both representations and planning strategies emerge from experience.

AI USE STATEMENT

The research ideas originated with the authors. We used generative AI tools primarily to refine the manuscript's language, assist with code development, and help run experiments. The authors reviewed all code and manuscript text. The authors compiled and organized the experimental results and take responsibility for the final manuscript, code, and reported findings, including all AI-assisted contributions.

REPRODUCIBILITY STATEMENT

Section C provides pseudocode. Section D specifies the model architectures, training settings, data collection, evaluation protocols, and baseline configurations for experiments.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

David Aldous and James Allen Fill. Reversible markov chains and random walks on graphs, 2002. Unfinished monograph, recompiled 2014, available at http://www.stat.berkeley.edu/aldous/RWG/book.html.

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

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

Katie C. Bittner, Aaron D. Milstein, Christine Grienberger, Sandro Romani, and Jeffrey C. Magee. Behavioral time scale synaptic plasticity underlies CA1 place fields. *Science*, 2017.

Rinu Boney, Juho Kannala, and Alexander Ilin. Regularizing model-based planning with energy-based models. In *CoRL*, 2020.

Krzysztof Burdzy and Zhen-Qing Chen. Discrete approximations to reflected Brownian motion. *The Annals of Probability*, 2008.

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.

Sheng Cheng, Deqian Kong, Jianwen Xie, Kookjin Lee, Ying Nian Wu, and Yezhou Yang. Latent space energy-based neural ODEs. *TMLR*, 2025.

Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 2006.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 1993.

Yilun Du, Toru Lin, and Igor Mordatch. Model-based planning with energy-based models. In *CoRL*, 2020.

Ben Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. In *NeurIPS*, 2019.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *NeurIPS*, 2022.

Benjamin Eysenbach, Vivek Myers, Ruslan Salakhutdinov, and Sergey Levine. Inference via interpolation: Contrastive representations provably enable planning and inference. In *NeurIPS*, 2024.

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

Nikolaos Gkanatsios, Ayush Jain, Zhou Xian, Yunchu Zhang, Christopher G. Atkeson, and Katerina Fragkiadaki. Energy-based models are zero-shot planners for compositional scene rearrangement. In *RSS*, 2023.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *NeurIPS*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML*, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *ICLR*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 2025.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *ICLR*, 2024.

Nicklas A. Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *ICML*, 2022.

Yanchen Huo, Ziying Song, and Yadan Luo. Flow-JEPA: Flow matching for robust latent dynamics in JEPA world models. *arXiv preprint arXiv:2608.29029*, 2026.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *ICML*, 2022.

Deqian Kong, Bo Pang, Tian Han, and Ying Nian Wu. Molecule design by latent space energy-based modeling and gradual distribution shifting. In *UAI*, 2023.

Deqian Kong, Yuhao Huang, Jianwen Xie, Edouardo Honig, Ming Xu, Shuanghong Xue, Pei Lin, Sanping Zhou, Sheng Zhong, Nanning Zheng, and Ying Nian Wu. Molecule design by latent prompt transformer. In *NeurIPS*, 2024a.

Deqian Kong, Dehong Xu, Minglu Zhao, Bo Pang, Jianwen Xie, Andrew Lizarraga, Yuhao Huang, Sirui Xie, and Ying Nian Wu. Latent plan transformer for trajectory abstraction: Planning as latent space inference. In *NeurIPS*, 2024b.

Deqian Kong, Minglu Zhao, Dehong Xu, Bo Pang, Shu Wang, Edouardo Honig, Zhangzhang Si, Chuan Li, Jianwen Xie, Sirui Xie, and Ying Nian Wu. Latent thought models with variational Bayes inference-time computation. In *ICML*, 2025.

Deqian Kong, Minglu Zhao, Aoyang Qin, Bo Pang, Chenxin Tao, David Hartmann, Edouardo Honig, Dehong Xu, Amit H. Kumar, Matthew Sarte, Chuan Li, Jianwen Xie, and Ying Nian Wu. Inference-time rethinking with latent thought vectors for math reasoning. In *Workshop on Latent & Implicit Thinking – Going Beyond CoT Reasoning, ICLR*, 2026.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *ICLR*, 2022.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Muyuan Liu, Yue Huang, Zheng Liang, and Xiang Gao. Toward physically grounded jepa world models for goal-conditioned robotic planning. *arXiv preprint arXiv:2609.03565*, 2026.

Yunhao Luo, Chen Sun, Joshua B. Tenenbaum, and Yilun Du. Potential based diffusion motion planning. In *ICML*, 2024.

Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *EMNLP*, 2018.

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

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *ICLR*, 2023.

Donghun Noh, Deqian Kong, Minglu Zhao, Andrew Lizarraga, Jianwen Xie, Ying Nian Wu, and Dennis Hong. Latent adaptive planner for dynamic manipulation. In *CoRL*, 2025.

James R Norris. Heat kernel asymptotics and the distance function in Lipschitz Riemannian manifolds. *Acta Mathematica*, 1997.

John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 1971.

Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. In *NeurIPS*, 2020.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline goal-conditioned RL with latent states as actions. In *NeurIPS*, 2023.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with Hilbert representations. In *ICML*, 2024.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *ICLR*, 2025.

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

Aoyang Qin, Deqian Kong, Wei Wang, Ying Nian Wu, Song-Chun Zhu, and Sirui Xie. Generative actor critic. *arXiv preprint arXiv:2512.21527*, 2025.

Rajkumar Vasudeva Raju, J Swaroop Guntupalli, Guangyao Zhou, Carter Wendelken, Miguel Lázaro-Gredilla, and Dileep George. Space is a latent sequence: A theory of the hippocampus. *Science Advances*, 2024.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied AI research. In *ICCV*, 2019.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 2020.

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

Stefan Steinerberger. Varadhan asymptotics for the heat kernel on finite graphs. *arXiv preprint arXiv:1801.02183*, 2018.

Guangyan Sun, Mingyu Jin, Zhenting Wang, Cheng-Long Wang, Siqi Ma, Qifan Wang, Tong Geng, Ying Nian Wu, Yongfeng Zhang, and Dongfang Liu. Visual agents as fast and slow thinkers. In *ICLR*, 2025.

Edward C Tolman. Cognitive maps in rats and men. *Psychological review*, 1948.

Edward C. Tolman and Charles H. Honzik. Introduction and removal of reward, and maze performance in rats. *University of California Publications in Psychology*, 1930.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In *NeurIPS*, 2021.

Julen Urain, An T. Le, Alexander Lambert, Georgia Chalvatzaki, Byron Boots, and Jan Peters. Learning implicit priors for motion optimization. In *IROS*, 2022.

Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *ICRA*, 2023.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Sathamangalam R Srinivasa Varadhan. On the behavior of the fundamental solution of the heat equation with variable coefficients. *Communications on Pure and Applied Mathematics*, 1967.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *ICML*, 2023.

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim G. J. Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning. In *ICML*, 2026.

Erik Wijmans, Manolis Savva, Irfan Essa, Stefan Lee, Ari S. Morcos, and Dhruv Batra. Emergence of maps in the memories of blind navigation agents. In *ICLR*, 2023.

Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. A theory of generative ConvNet. In *ICML*, 2016.

Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. Cooperative learning of energy-based model and latent variable model via MCMC teaching. In *AAAI*, 2018.

Dehong Xu, Ruiqi Gao, Wen-Hao Zhang, Xue-Xin Wei, and Ying Nian Wu. On conformal isometry of grid cells: Learning distance-preserving position embedding. In *ICLR*, 2025.

Yan Xu, Deqian Kong, Dehong Xu, Ziwei Ji, Bo Pang, Pascale Fung, and Ying Nian Wu. Diverse and faithful knowledge-grounded dialogue generation via sequential posterior inference. In *ICML*, 2023a.

Yifei Xu, Jianwen Xie, Tianyang Zhao, Chris Baker, Yibiao Zhao, and Ying Nian Wu. Energy-based continuous inverse optimal control. *IEEE Transactions on Neural Networks and Learning Systems*, 2023b.

Zhutian Yang, Jiayuan Mao, Yilun Du, Jiajun Wu, Joshua B. Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Compositional diffusion-based continuous constraint solvers. In *CoRL*, 2023.

Peiyu Yu, Dinghuai Zhang, Hengzhi He, Xiaojian Ma, Sirui Xie, Ruiyao Miao, Yifan Lu, Yasi Zhang, Deqian Kong, Ruiqi Gao, Jianwen Xie, Guang Cheng, and Ying Nian Wu. "noisier" noise contrastive estimation is (almost) maximum likelihood. In *ICLR*, 2026a.

Zhoushun Yu, Xiaoyu Hu, and Xiangyu Xu. QQWorld: Quantile-quantile matching for world model regularization. *arXiv preprint arXiv:2607.28415*, 2026b.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

Minglu Zhao, Dehong Xu, Deqian Kong, Wen-Hao Zhang, and Ying Nian Wu. A minimalistic representation model for head direction system. In *Proceedings of the 47th Annual Conference of the Cognitive Science Society*, 2025a.

Minglu Zhao, Dehong Xu, Deqian Kong, Wen-Hao Zhang, and Ying Nian Wu. Place cells as multi-scale position embeddings: Random walk transition kernels for path planning. In *NeurIPS*, 2025b.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *ICML*, 2025.

Alicja Ziarko, Michał Bortkiewicz, Michał Zawalski, Benjamin Eysenbach, and Piotr Miłoś. Contrastive representations for temporal reasoning. In *NeurIPS*, 2025.

APPENDIX

A Extended related work **16**

B Derivations and Proofs **17**

B.1 Density-ratio learning and normalization 17

B.2 Short horizons: geodesic geometry and spatial scale 17

B.3 Long horizons: connectivity and mixing 18

B.4 What the embedding parameterization preserves 19

B.5 Planning: GP and PAP as temperature limits 20

B.6 Geometric weights and goal reaching 21

C Algorithms **22**

D Experiments **24**

D.1 Representation Probes 24

D.2 Demonstrations 25

D.3 Additional Ablation 26

D.4 Models, Data and Training setting 27

D.5 Evaluation Protocols 29

A Extended related work

Latent world models. Latent world models predict how a latent state evolves under actions, rather than predicting future frames, and plan or learn policies in that latent state space. They can be grouped by the training signal that shapes the latent state. Reconstruction-based models learn the latent through a decoder that reconstructs observations (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; 2025; Micheli et al., 2023). Reconstruction provides a dense training signal, but it pushes the latent to encode visual detail regardless of its relevance to control. Reward-driven models remove the decoder and shape the latent through task signals such as reward and value prediction, often combined with a latent consistency loss (Schrittwieser et al., 2020; Hansen et al., 2022; 2024). These objectives encourage the latent to retain information useful for reward prediction and control, but require reward labels and tie representation learning to the training tasks. JEPA-style world models need neither a decoder nor rewards: they are trained to predict future embeddings from reward-free trajectories, and they plan by reaching a goal embedding. With no reconstruction or reward signal to anchor the latent, their central challenge is representation collapse. One line of work avoids it by freezing a pretrained visual encoder during action-conditioned dynamics training, as in DINO-WM and V-JEPA 2-AC (Zhou et al., 2025; Assran et al., 2025), while another trains the encoder end-to-end with explicit anti-collapse regularization (Sobal et al., 2025; Maes et al., 2026). LeWorldModel uses a single Gaussian regularizer (Maes et al., 2026). Subsequent work studies alternative regularization (Yu et al., 2026b), dynamics models (Huo et al., 2026; Gao & Xu, 2026), auxiliary physical supervision (Liu et al., 2026), hierarchical planning (Caselli et al., 2026), and action optimization with a terminal state-space energy over a frozen world model (Pham & Bera, 2026).

Energy-based models and inference for planning. Energy-based models (EBMs) learn distributions through energy functions, with foundational work connecting generative ConvNets to analysis-by-synthesis learning and cooperative training with latent generators (Xie et al., 2016; 2018). Noise-contrastive estimation learns density ratios from data and reference samples (Gutmann & Hyvärinen, 2010; Yu et al., 2026a). For control, learned energies serve as costs in inverse optimal control (Xu et al., 2023b), guide state-sequence planning (Du et al., 2020), and regularize planning toward likely transitions (Boney et al., 2020). Energy-based motion planners and diffusion planners further support trajectory optimization and compositional conditioning (Urain et al., 2022; 2023; Luo et al., 2024; Gkanatsios et al., 2023; Yang et al., 2023; Janner et al., 2022; Ajay et al., 2022). A related line moves inference into latent spaces, using latent plans for return-conditioned generation, online adaptation, and policy improvement (Kong et al., 2024b; Noh et al., 2025; Qin et al., 2025), or latent EBMs for task adaptation and continuous-time dynamics (Pang et al., 2020; Cheng et al., 2025). Beyond control, latent inference supports molecular design (Kong et al., 2023; 2024a), knowledge-grounded dialogue (Xu et al., 2023a), and reasoning through latent thoughts and iterative refinement (Kong et al., 2025; 2026). FaST studies a complementary form of adaptive computation, switching between fast responses and slower inference for visual reasoning (Sun et al., 2025). Spatial representation models connect navigation to head-direction coding (Zhao et al., 2025a), grid-cell embeddings that preserve local distances up to a scale factor (Xu et al., 2025), and place-cell embeddings of multiscale random-walk kernels (Zhao et al., 2025b). Our model learns conditional temporal log-density ratios from observation pairs through NCE. These scores provide long-range goal guidance for local action evaluation. GP retains the joint move-and-horizon maximization of Zhao et al. (2025b), applied to estimated density-ratio progress, while our temperature objective also yields PAP, which aggregates progress across horizons into a single goal-dependent potential.

B Derivations and Proofs

We first show what NCE learns from temporal pairs, then derive the geometric information in these relations at short and long horizons. We next examine how the embedding model represents this information and how GP and PAP use it to evaluate progress toward a goal.

The geometric results concern the ideal score $G^* = \log[p_{\mathrm{data}}(y|x,\tau)/p_0(y)]$ under the stated diffusion or Markov assumptions. For the Markov analysis to apply directly to observations, each observation must determine the state. These results describe the ideal temporal relations; learning at finitely many horizons does not by itself establish the same limiting behavior for G_θ. The planning identities, however, hold for any finite learned scores.

B.1 Density-ratio learning and normalization

NCE learns temporal association by distinguishing observed pairs from pairs with independently sampled targets. Assume that the reference p_0 covers the conditional target support. At a sampled horizon, the two pair densities are $p_{\mathrm{data}}(x, y|\tau)$ and $p_{\mathrm{data}}(x|\tau)p_0(y)$. With N negatives per positive, the class prior odds are $1 : N$. The source marginal cancels from the posterior odds, leaving $p_{\mathrm{data}}(y|x,\tau)/(Np_0(y))$.

The classifier $\sigma(G - \log N)$ has odds e^G/N. Matching these odds gives the unrestricted population minimizer of the logistic loss in Eq. (5):

$$G^*(x, y, \tau) = \log \frac{p_{\mathrm{data}}(y|x,\tau)}{p_0(y)}.$$

This equality holds on sampled sources and horizons where both densities are positive (Gutmann & Hyvärinen, 2010; Ma & Collins, 2018). Where $p_0(y) > 0$ and the data conditional is zero, the optimum is approached as $G \to -\infty$, and we interpret $e^{G^*} = 0$.

The fitted score need not satisfy the same normalization as this ideal ratio. For the normalized EBM in Eq. (1), the exact relationship is

$$\log \frac{p_\theta(y|x,\tau)}{p_0(y)} = G_\theta(x, y, \tau) - \log \mathbb{E}_{p_0(y')}\left[e^{G_\theta(x,y',\tau)}\right]. \tag{12}$$

The expectation equals $e^{\beta_1(\tau)}Z_\theta(x,\tau)$ and measures how far the fitted ratio is from integrating to one. The ideal ratio integrates to one at every source. A shared offset $\beta_1(\tau)$ cannot enforce this when $Z_\theta(x,\tau)$ varies with x. This matters during planning: candidates are different sources paired with the same goal, so a source-dependent correction can change their ranking. The planner uses the fitted score G_θ.

B.2 Short horizons: geodesic geometry and spatial scale

Short temporal relations reveal path geometry through the diffusion limit of the walk. To obtain this limit, shrink free-space steps by δ and assign each step time δ^2. The covariance per unit time remains $2\alpha I_n$. Under the invariance principle assumed in Sec. 3, the walk converges to reflected Brownian motion with generator $\alpha\Delta$ (Burdzy & Chen, 2008). Reflection prevents probability from flowing through walls. We first take this continuum limit, then let diffusion time tend to zero.

Proof of Proposition 1. On the smooth, bounded, connected domain of the proposition, Varadhan's formula for the reflecting heat kernel gives, for fixed interior x, y,

$$-4\alpha\tau G^*(x, y, \tau) = -4\alpha\tau \log p_{\mathrm{data}}(y|x,\tau) + 4\alpha\tau \log p_0(y) \longrightarrow d_{\mathrm{geo}}(x, y)^2,$$

because the uniform reference is positive and independent of τ (Varadhan, 1967; Norris, 1997). □

This limit also explains how the score ranks candidate moves. Among a finite set of interior candidates, suppose one has strictly shorter geodesic distance to y_g than all the others. It maximizes $G^*(\cdot, y_g, \tau)$ for sufficiently small τ. Indeed, its score advantage over each competitor, multiplied by $4\alpha\tau$, tends to the strictly positive difference of squared distances.

The order of limits matters. On a fixed graph, let $\ell \geq 1$ be the smallest number of edges between distinct reachable states x, y. Then $P^k(x, y) = 0$ for $k < \ell$. Even the continuous-time walk on this graph satisfies

$$[e^{\tau(P-I)}](x, y) = \tau^\ell P^\ell(x, y)/\ell! + O(\tau^{\ell+1}).$$

Its logarithm therefore scales as $\ell \log \tau$, rather than $-d^2_{\text{geo}}/(4\alpha\tau)$ (Steinerberger, 2018).

The horizon giving the strongest local progress. The geodesic limit describes which candidate scores higher. To see which horizon gives the strongest progress signal, consider diffusion in free space $\mathbb{R}^n$, whose transition density is

$$p_{\text{data}}(y|x, \tau) = (4\pi\alpha\tau)^{-n/2} \exp\left(-\frac{\|y - x\|_2^2}{4\alpha\tau}\right). \tag{13}$$

Fix $x \neq y_g$, and write the goal distance as $r = \|y_g - x\|_2$ and the direction toward it as $u = (y_g - x)/r$. For a fixed reference with $0 < p_0(y_g) < \infty$, the change in the ratio per unit step toward the goal is

$$\frac{\partial}{\partial\epsilon} e^{G^*(x+\epsilon u, y_g, \tau)}\Big|_{\epsilon=0} = \frac{r}{2\alpha\tau} \frac{p_{\text{data}}(y_g|x, \tau)}{p_0(y_g)}. \tag{14}$$

We maximize this local progress over the horizon. Its logarithmic derivative in τ is $-(n + 2)/(2\tau) + r^2/(4\alpha\tau^2)$, which changes from positive to negative at

$$\tau^* = \frac{r^2}{2(n + 2)\alpha}, \qquad \sqrt{2n\alpha\tau^*} = r\sqrt{\frac{n}{n + 2}}. \tag{15}$$

The strongest local signal therefore occurs when the walk's root-mean-square displacement is a fixed fraction of the goal distance. At shorter horizons, little probability has reached the goal; at longer horizons, probability has spread over a larger region. Their balance gives the squared-distance scaling in Sec. 3. This calculation concerns infinitesimal motion in free space; finite moves, walls, and passages can change the maximizing horizon.

B.3 Long horizons: connectivity and mixing

At longer horizons, the walk loses information about its starting position. The spectral expansion describes which differences persist as this happens. We now return to a finite state space $\mathcal{S}$ and a discrete walk with irreducible, aperiodic, reversible transition matrix P. Let π be its stationary distribution, set $p_0 = \pi$, and order the eigenvalues as $1 = \lambda_1 > \lambda_2 \geq \cdots$. At integer horizons, $p_{\text{data}}(y|x, \tau) = P^\tau(x, y)$.

Proof of Proposition 2. Put $D_\pi = \text{diag}(\pi)$. Detailed balance makes $S = D_\pi^{1/2} P D_\pi^{-1/2}$ symmetric. If $S\phi_j = \lambda_j\phi_j$ with orthonormal ϕ_j, then $\psi_j = D_\pi^{-1/2}\phi_j$ are orthonormal under π. Irreducibility gives $\lambda_1 = 1$, $\psi_1 = 1$; aperiodicity gives $|\lambda_j| < 1$ for $j \geq 2$. Expanding $P^\tau = D_\pi^{-1/2} S^\tau D_\pi^{1/2}$ yields

$$e^{G^*(x,y,\tau)} = \frac{P^\tau(x, y)}{\pi(y)} = 1 + \sum_{j\geq 2} \lambda_j^\tau \psi_j(x)\psi_j(y). \tag{16}$$

Taking logarithms where $P^\tau(x, y) > 0$ proves the proposition (Aldous & Fill, 2002; Coifman & Lafon, 2006); the ratio identity also holds at zero transition probability. □

Why weak connections produce slow modes. To connect the spectrum to the environment, we measure how much a function changes across transitions of the walk. Write $\langle f, g\rangle_\pi = \sum_x \pi(x) f(x) g(x)$. Expanding the squared change and using stationarity gives

$$\begin{aligned} \mathcal{E}(f) &:= \frac{1}{2} \sum_{u,v} \pi(u) P(u, v)[f(u) - f(v)]^2 = \langle f, (I - P)f\rangle_\pi, \\ \mathcal{E}(\psi_j) &= 1 - \lambda_j. \end{aligned} \tag{17}$$

An eigenfunction with eigenvalue near one therefore varies little across transitions, on average. Now divide the states into regions A and A^c, with $0 < \pi(A) < 1$. The function $f = (\mathbf{1}_A -$

$\pi(A))/\sqrt{\pi(A)\pi(A^c)}$ is constant within each region and has zero mean and unit norm under π. Substituting it into the variational formula $1-\lambda_2 = \min_{\langle f,1\rangle_\pi=0,\,\langle f,f\rangle_\pi=1} \mathcal{E}(f)$ gives

$$1-\lambda_2 \le \frac{Q(A,A^c)}{\pi(A)\pi(A^c)}, \qquad Q(A,A^c) = \sum_{u\in A,\, v\notin A} \pi(u)P(u,v). \tag{18}$$

Here $Q(A,A^c)$ is the stationary probability flow from one region to the other. Only transitions between the regions contribute to $\mathcal{E}(f)$, and reversibility makes the two directions equal (Aldous & Fill, 2002, Sec. 3.6). When this flow is small relative to the regions' stationary masses, the bound places λ_2 near one.

For two rooms joined by a narrow doorway, this gives a mode that decays slowly. A large change across the rarely crossed doorway contributes little to its Dirichlet energy, while changes across frequent transitions within a room contribute more. When each room mixes quickly, this mode describes a contrast between the rooms, as in Sec. 3.

What survives before mixing. Eventually, mixing removes these differences between starting states. To bound the remaining contrast, let $\rho = \max_{j\ge 2}|\lambda_j| < 1$. Completeness gives $\sum_{j\ge 2}\psi_j(x)^2 = \pi(x)^{-1}-1$, so Cauchy–Schwarz in Eq. (16) yields

$$\left|e^{G^*(x,y,\tau)}-1\right| \le \rho^\tau \sqrt{\big(\pi(x)^{-1}-1\big)\big(\pi(y)^{-1}-1\big)}. \tag{19}$$

Thus every ratio approaches one, $G^*\to 0$, and progress between any two sources vanishes. When one positive mode decays more slowly than all the others, it determines the leading contrast. Specifically, if $\lambda_2 > 0$ and $\lambda_2 > |\lambda_j|$ for every $j \ge 3$, then applying $\log(1+z) = z + O(z^2)$ to Eq. (16) gives

$$G^*(x,y,\tau) = \lambda_2^\tau \psi_2(x)\psi_2(y) + o(\lambda_2^\tau). \tag{20}$$

Without this separation, we must retain all dominant modes in the full expansion. Negative eigenvalues contribute terms that alternate in sign between odd and even horizons.

B.4 What the embedding parameterization preserves

The learned score describes temporal relations through embedding alignment. When both embeddings have unit norm, we can express the same relation through their distance:

$$e^{G_\theta(x,y,\tau)} = e^{\beta_0(\tau)+\beta_1(\tau)} \exp\left[-\frac{\beta_0(\tau)}{2}\|h_\theta(x,\tau)-g_\theta(y,\tau)\|_2^2\right]. \tag{21}$$

This identity holds for any learned coefficient $\beta_0(\tau)$ and offset $\beta_1(\tau)$. The coefficient controls how the score changes with embedding distance; its sign determines the direction of this dependence. At a fixed horizon, the offset rescales all exponentiated scores equally and leaves their ordering unchanged. Across horizons, these scales affect how much each horizon contributes to planning.

For tied encoders, comparison with the goal itself gives the exact identity

$$G_\theta(y_g,y_g,\tau) - G_\theta(x,y_g,\tau) = \frac{\beta_0(\tau)}{2}\|h_\theta(x,\tau)-h_\theta(y_g,\tau)\|_2^2. \tag{22}$$

In our evaluated tied-encoder models, the fitted alignment coefficients are positive and the offsets are negative. The positive coefficients make closer embeddings score higher and place the goal at a score maximum at each evaluated horizon, consistent with the geometric interpretation. This is an observation about the fitted models. A maximum at the goal does not rule out other local maxima or ensure that an improving move is available at every state.

Tied encoders also preserve a structural property of reversible walks. At a fixed horizon on a finite state space with $p_0(x) > 0$, suppress the horizon argument and write $K_{xy} = \exp(\beta_0\langle h_\theta(x), h_\theta(y)\rangle)$. This kernel is symmetric for either sign of β_0. The normalized conditional matrix $p_0(y)K_{xy}/Z_\theta(x)$ is therefore reversible: with stationary weights proportional to $p_0(x)Z_\theta(x)$, the flow between x and y is proportional to $p_0(x)p_0(y)K_{xy}$, which is symmetric. This holds at each horizon, but does not require the learned conditionals to be powers of a single transition matrix.

These properties do not establish exact representation of every reversible walk. Tied unit embeddings give the same diagonal score at every state, whereas the walk's return ratio can vary from state to state.

There is a related feature construction for the ratio itself. For integers $k \geq 1$, define the nonnegative features $f_k(x)_z = P^k(x,z)/\sqrt{\pi(z)}$. Detailed balance and matrix multiplication give $\langle f_k(x), f_k(y)\rangle = P^{2k}(x,y)/\pi(y)$. This diffusion-kernel construction (Coifman & Lafon, 2006) represents the ratio as an inner product. Our affine inner product models its logarithm, so the construction does not establish exact representation by our model.

B.5 Planning: GP and PAP as temperature limits

The same action can improve the goal score at one horizon and reduce it at another. The temperature objective lets each action place more weight on horizons where it makes progress, while penalizing departures from uniform weights.

Fix a goal y_g, a nonempty finite horizon set $\mathcal{T}$, and finite scores. For a candidate action a, write $\Delta_\tau(a) = e^{G_\theta(\widehat{x}^a_{t+1}, y_g, \tau)} - e^{G_\theta(x_t, y_g, \tau)}$. Let U be the uniform distribution on $\mathcal{T}$, and let $\mathcal{P}(\mathcal{T})$ be the set of distributions over these horizons.

Proposition 3 (GP and PAP as temperature limits)**.** *For $\beta > 0$, the planning objective satisfies*

$$\begin{aligned} V_\beta(a) &:= \beta \log\left[\frac{1}{|\mathcal{T}|}\sum_\tau e^{\Delta_\tau(a)/\beta}\right] \\ &= \max_{q\in\mathcal{P}(\mathcal{T})}\left\{\sum_\tau q(\tau)\Delta_\tau(a) - \beta\,\mathrm{KL}(q\|U)\right\}, \end{aligned} \tag{23}$$

with unique maximizer $q_\beta(\tau|a) \propto e^{\Delta_\tau(a)/\beta}$. As $\beta \to 0^+$, $V_\beta(a)$ converges to $\max_\tau \Delta_\tau(a)$ (GP); as $\beta \to \infty$ it converges to $|\mathcal{T}|^{-1}\sum_\tau \Delta_\tau(a)$ (PAP).

Proof. With $\mathrm{KL}(q\|U) = \sum_\tau q(\tau)\log(|\mathcal{T}|q(\tau))$ and $0\log 0 = 0$, direct substitution gives

$$\sum_\tau q(\tau)\Delta_\tau(a) - \beta\,\mathrm{KL}(q\|U) = V_\beta(a) - \beta\,\mathrm{KL}(q\|q_\beta(\cdot|a)). \tag{24}$$

The right-hand side is largest only when $q = q_\beta$, because KL is nonnegative and vanishes only when the two distributions agree. This proves the variational identity and uniqueness. To obtain the limits, write $M = \max_\tau \Delta_\tau(a)$. Evaluating the variational objective at weights concentrated on a maximizing horizon, at U, and at q_β gives

$$\begin{aligned} &M - \beta\log|\mathcal{T}| \leq V_\beta(a) \leq M, \\ &\frac{1}{|\mathcal{T}|}\sum_\tau \Delta_\tau(a) \leq V_\beta(a) \leq \sum_\tau q_\beta(\tau|a)\Delta_\tau(a). \end{aligned} \tag{25}$$

The first line gives the GP limit; the second gives the PAP limit because $q_\beta \to U$ as $\beta \to \infty$. □

At low temperature, an action is judged by the horizon where it makes the most progress. At high temperature, all horizons receive equal weight, so gains and losses offset each other. Multiplying all progress values by the same positive constant preserves GP and PAP rankings. At finite temperature, preserving the ranking also requires scaling β by that constant.

Potential ascent and revisits. PAP evaluates every move through one function of the state. For the fixed goal, define

$$\Phi(x) = \frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}} e^{G_\theta(x, y_g, \tau)}. \tag{26}$$

PAP progress is exactly $\Phi(\widehat{x}^a_{t+1}) - \Phi(x_t)$. Since the current-state term is shared by all actions, PAP ranks candidates by their next-state potential. If the model, goal, and horizon set stay fixed, predictions match deterministic outcomes, and every executed step has strictly positive progress, then $\Phi(x_{t+1}) > \Phi(x_t)$. Returning to a previous state would restore its previous potential, which is impossible. The argument does not ensure that an improving action exists at every non-goal state.

GP can favor different horizons on successive moves and need not increase a common state potential. Consider two non-goal states with exponentiated score vectors $(0.2, 0.8)$ and $(0.8, 0.2)$ across two

horizons. Each state offers a move to the other and a stay action. Moving improves one component by 0.6 and reduces the other by 0.6, so GP prefers moving in both directions and cycles between the states. At every finite temperature, moving also gives $V_\beta = \beta \log \cosh(0.6/\beta) > 0$, while staying gives zero. PAP assigns both states potential 0.5, so neither move has strictly positive PAP progress. This cycling example is realizable even with tied, positive unit embeddings,[1] showing that these embedding constraints do not prevent GP from cycling. The scores need not be calibrated temporal ratios of a common walk, so the example concerns the learned-score planner.

B.6 Geometric weights and goal reaching

With exact transition ratios, the potential has a direct interpretation under the exploration walk: $|\mathcal{T}| p_0(y_g)\Phi(x)$ is the expected number of visits to y_g at horizons in $\mathcal{T}$. Counting visits differs from measuring whether the goal is reached, since one trajectory may visit it several times.

Geometric weights let us relate these two quantities. Consider any finite Markov chain P, take a fixed $p_0(y_g) > 0$, and let $0 < \gamma < 1$. Weighting all horizons, including zero, defines

$$\Phi_\gamma(x) = (1-\gamma) \sum_{\tau=0}^{\infty} \gamma^\tau \frac{P^\tau(x, y_g)}{p_0(y_g)}. \tag{27}$$

This potential measures discounted occupancy of the goal relative to $p_0(y_g)$. The following result relates it to the time of first arrival. Neither reversibility nor an absorbing goal is required.

Proposition 4 (Geometric weights connect occupancy to first arrival)**.** *Let $T_g = \inf\{t \geq 0 : X_t = y_g\}$ be the first arrival time under P, with $\gamma^\infty = 0$, and let $\mathbb{E}_x$ denote expectation for the walk started at x. Then*

$$\Phi_\gamma(x) = \mathbb{E}_x[\gamma^{T_g}]\, \Phi_\gamma(y_g). \tag{28}$$

Every non-goal state from which the goal is reachable satisfies

$$\max_{y : P(x,y) > 0} \Phi_\gamma(y) \geq \Phi_\gamma(x)/\gamma > \Phi_\gamma(x). \tag{29}$$

Proof. There are no visits to the goal before T_g. Once the walk arrives, its subsequent visits have the same law as those of a walk started at the goal. The strong Markov property therefore gives

$$\mathbb{E}_x\left[\sum_{t \geq 0} \gamma^t \mathbf{1}\{X_t = y_g\}\right] = \mathbb{E}_x[\gamma^{T_g}]\, \mathbb{E}_{y_g}\left[\sum_{j \geq 0} \gamma^j \mathbf{1}\{X_j = y_g\}\right].$$

Multiplying by $(1-\gamma)/p_0(y_g)$ proves Eq. (28) (Aldous & Fill, 2002, Lemma 2.25). To show that an improving move exists, split off the horizon-zero term: $\Phi_\gamma(x) = \frac{1-\gamma}{p_0(y_g)} \mathbf{1}\{x = y_g\} + \gamma \sum_y P(x,y)\Phi_\gamma(y)$. At a reachable non-goal state, $\Phi_\gamma(x) > 0$ and the indicator vanishes. The average next-state potential is therefore $\Phi_\gamma(x)/\gamma > \Phi_\gamma(x)$. At least one successor with $P(x,y) > 0$ has value at least this average, proving Eq. (29). □

This gives a goal-reaching guarantee under exact local control. Suppose all walk-supported moves are available as deterministic actions, their outcomes are predicted exactly, and planning stops at the goal. From any start that can reach the goal under P, choosing a successor with largest Φ_γ strictly increases the potential. Its value remains positive, so the goal remains reachable. No state can be revisited, and the trajectory cannot stop at a non-goal state. It must therefore reach the goal in at most $|\mathcal{S}| - 1$ moves. The known horizon-zero term anchors the goal value; omitting it can destroy this guarantee.

The geometric aggregate is proportional to discounted occupancy, the quantity connected to contrastive goal-conditioned RL (Eysenbach et al., 2022). Dividing by its value at the goal gives the discounted first-arrival value. Under the exact deterministic control assumptions above, choosing the successor with largest value is a policy-improvement step over the exploration behavior. Our planner uses a finite set of uniformly weighted positive horizons, which does not generally satisfy this Bellman identity. The geometric extension therefore explains a connection to goal reaching.

[1] Choose positive unit vectors u, v with $\langle u, v \rangle = 1/2$, use u for the goal at both horizons, and assign (v, u) and (u, v) to the two states across horizons. Setting $\beta_0 = \log 16$ and $\beta_1 = \log 0.05$ gives the stated exponentiated scores.

C ALGORITHMS

Algorithms 1 and 2 learn the temporal representation and local dynamics separately. Algorithm 3 uses both models for closed-loop planning, with their parameters fixed.

Algorithm 1: Learning θ

Input: Exploration trajectories, $\mathcal{T}$, $p(\tau)$, reference sampler p_0, and negative count N.

1. Initialize temporal parameters θ.
2. **while** temporal updates remain **do**
3. Sample a horizon $\tau \sim p(\tau)$.
4. Sample a batch of pairs (x, y) from $p_{\text{data}}(x, y|\tau)$ within episodes.
5. For each source x, sample N independent negative targets from p_0.
6. Compute $G_\theta(x, y, \tau) - \log N$ for positive and negative pairs.
7. Update θ by Eq. (5): average the positive loss plus the sum of negative losses over sources.
8. **end while**
9. **return** learned parameters θ.

Algorithm 2: Learning ϕ

Input: Action-labeled one-step transitions with distribution p_{step}.

1. Initialize local-dynamics parameters ϕ.
2. **while** dynamics updates remain **do**
3. Sample a batch of transitions $(x_t, a_t, x_{t+1}) \sim p_{\text{step}}$.
4. Update ϕ by Eq. (6) for state prediction.
5. **end while**
6. **return** learned parameters ϕ.

Algorithm 3: Planning

Input: Frozen G_θ and F_ϕ, goal y_g, horizon set $\mathcal{T}$, temperature $\beta > 0$ (or either limit), and action budget.

1. Observe the initial x_0 and set $t = 0$.
2. **while** goal not reached and budget remains **do**
3. Construct candidate set $\mathcal{A}(x_t)$.
4. **for each** $a \in \mathcal{A}(x_t)$ **do**
5. Predict $\widehat{x}^a_{t+1} = F_\phi(x_t, a)$.
6. Compute progress for each $\tau \in \mathcal{T}$: $\Delta_\tau(a) = e^{G_\theta(\widehat{x}^a_{t+1}, y_g, \tau)} - e^{G_\theta(x_t, y_g, \tau)}$.
7. **end for**
8. Select a^*_t by Eq. (8), using maximum progress as $\beta \to 0^+$ or mean progress as $\beta \to \infty$.
9. Execute only a^*_t in the environment.
10. Receive the actual x_{t+1} and set $t \leftarrow t + 1$.
11. **end while**

PyTorch-Style Pseudocode

The compact version below shows one update for each model and one planning decision. `pair_score` returns a $B \times B$ matrix of temporal scores at a shared lag; `goal_score` returns candidate-by-horizon scores. Here `F` denotes `torch.nn.functional`. Planning uses every integer horizon from 1 to $\tau_{\max}$.

```
def train_temporal(current, future, tau):
    logits = temporal.pair_score(current, future, tau)
    logits = logits - math.log(len(current) - 1)
    positives = logits.diagonal()
    negatives = F.softplus(logits).sum(1)
    negatives = negatives - F.softplus(positives)
    loss = (F.softplus(-positives) + negatives).mean()
    temporal_opt.zero_grad()
    loss.backward()
    temporal_opt.step()

def train_dynamics(state, action, next_state):
    prediction = dynamics(state, action)
    error = (prediction - next_state).square()
    loss = error.flatten(1).sum(1).mean()
    dynamics_opt.zero_grad()
    loss.backward()
    dynamics_opt.step()

@torch.no_grad()
def choose_action(state, goal, actions, taus, planner):
    states = state.expand(len(actions), -1)
    next_states = dynamics(states, actions)
    here = temporal.goal_score(state, goal, taus).exp()
    there = temporal.goal_score(next_states, goal, taus).exp()
    progress = there - here
    if planner == "GP":
        value = progress.amax(1)
    else:
        value = progress.mean(1)
    return actions[value.argmax()]
```

At test time, both models are in evaluation mode. Execute the returned action, observe the new state, and repeat. The dynamics call illustrates learned state prediction; oracle experiments instead obtain candidate outcomes from the simulator, as specified in Sec. D.

D EXPERIMENTS

D.1 REPRESENTATION PROBES

Representation Probes and Visualizations. Figure 4 and Tab. 4 compare embedding distances with maze geometry at different horizons. Near pairs are at most eight geodesic steps apart. Each t-SNE panel is fitted separately and aligned to the maze by an orthogonal Procrustes transform; projection coordinates are not physical distances.

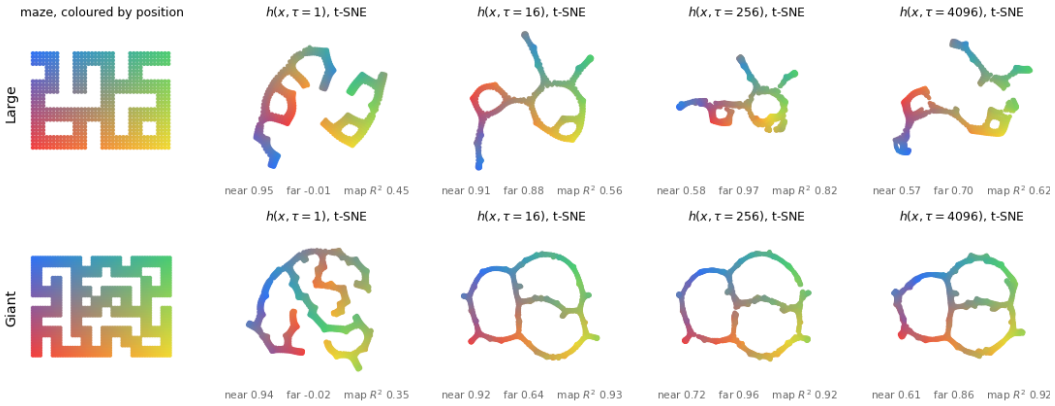

Figure 4: Learned embeddings $h(x, \tau)$ on the Large (top) and Giant (bottom) state mazes; the left column colours each cell by its position and the other columns keep those colours. Every panel is rotated and reflected onto the maze by Procrustes, since t-SNE leaves the orientation undetermined.

Table 4: Spearman correlation between embedding distance and geodesic distance for near (≤ 8 steps) and far pairs, Large maze.

τ	1	16	256	4096
near	**0.95**	0.91	0.58	0.57
far	−0.01	0.88	**0.97**	0.70

Three-dimensional views. Figure 5 supplements the Giant-maze 2D projections in Fig. 1B with 3D t-SNE views of both mazes at $\tau = 1$, 16, and 4096. We use perplexity 30, PCA initialization, and random seed 0, fitting each maze and horizon separately. Rotation/reflection and uniform scaling align each fit to the maze plane without flattening its third dimension. Within each maze, all three panels share a camera angle and coordinate scale. The projections illustrate neighbourhood structure, not physical distances or comparable coordinates across horizons.

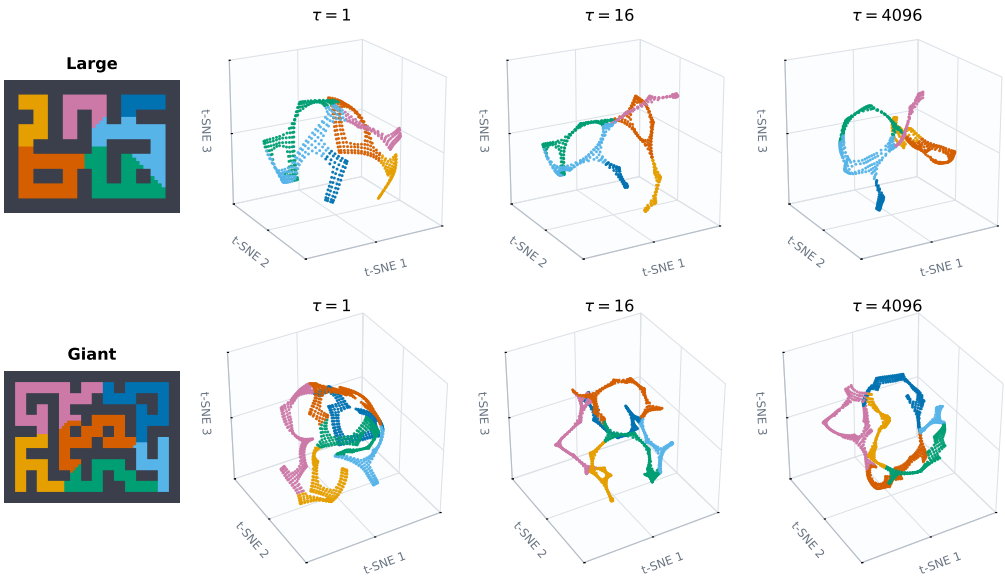

Figure 5: 3D t-SNE of state embeddings on Large (top) and Giant (bottom). At $\tau = 1$, 3D views reveal connectivity obscured in 2D. Colours match the maze regions at left; axes are not physical coordinates.

D.2 Demonstrations

Planning demonstrations. Figure 1C uses all integer horizons from 1 to 512 on Large (tasks 1 and 3) and from 1 to 2048 on Giant (tasks 1 and 2), with one-step lookahead and execution (0.2 units). These illustrative rollouts use different horizon ranges and a different planning radius from the aggregate evaluations in Tab. 1. For GP, path colour marks the horizon with the largest improvement for the chosen action; for PAP, it marks the horizon contributing most to the chosen candidate's pooled score, not a separately selected planning horizon. The recorded values are not temporally smoothed. Matching marker colours pair each start with its recorded episode goal.

Manipulation demonstrations. Figure 6 shows three planned episodes of different lengths, with evenly spaced frames followed by the goal image.

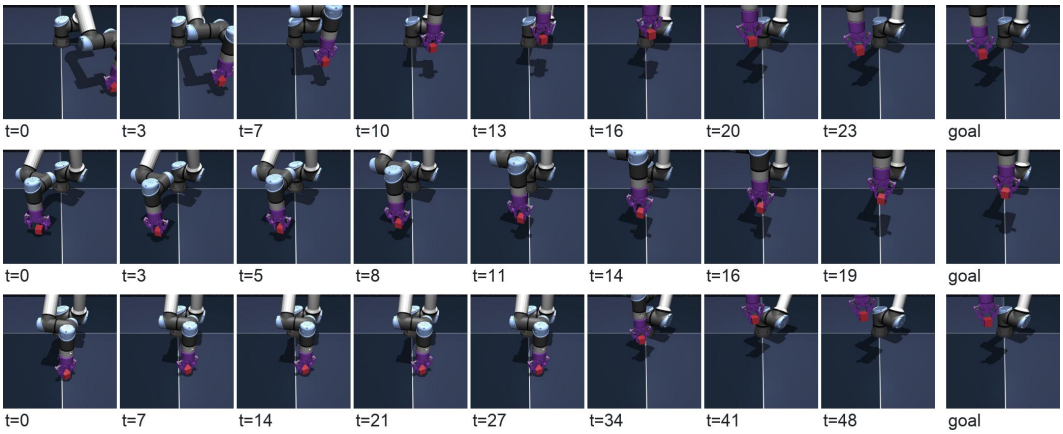

Figure 6: Three held-out cube-single episodes planned from pixels, one per row, each showing eight evenly spaced executed steps with the goal image last. The cube travels 0.455, 0.277, and 0.325 m, respectively; in every row the planner reaches the cube, grasps it, and carries it to the goal.

D.3 Additional Ablation

Learned Local Dynamics. The planner's local model predicts the next position as $x' = x + 0.2\,a + g_\phi(x, a)$, where g_ϕ is an MLP on Fourier features of the position and the action (four layers of width 512 on Large; six layers of width 1024 on Giant). It is trained by squared error on random-action data (1M transitions on Large; 10M transitions on Giant, since the 1M set leaves some wall segments with too few contacts to learn), with wall-contact steps weighted five times (Large) or twenty times (Giant). Held-out one-step error is below 0.001 units in both mazes, and the 15-step endpoint error against the simulator is 0.011 and 0.047 units, with the reachability decision of Tab. 13 agreeing with the simulator on 99.5% of candidates in both mazes. A Giant model trained on 40M transitions reaches a 15-step error of 0.008 units and full agreement; the tables use the 10M model, which already reproduces the probed results. Table 5 compares planning through the learned model with planning from simulator probes: the two agree within seed variation.

Table 5: Planning through the learned local dynamics model against simulator probes, $\tau_{\max} = 8192$. SR with SPL in parentheses.

		Official tasks		Random pairs	
Model	Planner	Learned	Probed	Learned	Probed
Large, state	PAP	0.75 (0.67)	0.70 (0.64)	0.90 (0.80)	0.95 (0.84)
Large, state	GP	0.96 (0.83)	1.00 (0.86)	0.95 (0.86)	0.95 (0.86)
Giant, state	PAP	0.88 (0.71)	0.87 (0.71)	0.80 (0.74)	0.90 (0.80)
Giant, state	GP	0.76 (0.58)	0.73 (0.57)	0.85 (0.76)	0.90 (0.80)
Large, images	PAP	0.58 (0.53)	0.60 (0.55)	0.95 (0.80)	1.00 (0.87)
Large, images	GP	0.84 (0.66)	0.85 (0.69)	0.95 (0.84)	0.95 (0.85)
Giant, images	PAP	0.36 (0.30)	0.36 (0.30)	0.70 (0.63)	0.70 (0.63)
Giant, images	GP	0.37 (0.24)	0.37 (0.24)	0.85 (0.70)	0.80 (0.65)

Types of Training Data Ablation. Table 6 trains every method on either random-action or lattice-walk data, separating the effect of the training data from that of the method.

Table 6: State planning with random-action or lattice data. Our local dynamics uses 1M random-action transitions on Large and 10M on Giant. SR averages three training seeds. The random-candidate control draws uniformly among the headings our planner scores, generated by the same learned local dynamics and filtered by the same back cone.

	Official tasks		**Random pairs**	
Method	Large	Giant	Large	Giant
State, random-action data (20 episodes of 50,000 steps)				
GCIQL (Kostrikov et al., 2022)	0.00	0.00	0.15	0.07
HIQL (Park et al., 2023)	0.00	0.00	0.12	0.10
HILP (Park et al., 2024)	0.49	0.00	0.47	0.18
PAP ($\beta \to \infty$)	0.59	0.06	0.70	0.38
GP ($\beta \to 0^+$)	0.90	0.04	0.80	0.38
State, lattice walk (one chain of 1M steps)				
GCIQL (Kostrikov et al., 2022)	0.00	0.00	0.72	0.22
HIQL (Park et al., 2023)	1.00	0.14	1.00	0.60
HILP (Park et al., 2024)	0.35	0.05	0.50	0.35
PAP ($\beta \to \infty$)	0.68	0.50	0.88	0.77
GP ($\beta \to 0^+$)	0.91	0.49	0.90	0.73
Random candidate (our candidate set)	0.06	0.02	0.58	0.45

Planning Radius. In Tab. 1, the default candidate radius is 3. We further explore the effect of different planning radius. We validated it afterwards on 50 held-out random pairs per maze, drawn with pair seed 1 at a geodesic distance of at least 20 cells and disjoint from the test pairs and the official tasks. The validation covers radii of 2, 3 and 4 units with both planners, $\tau_{\max} = 8192$, a budget of 5,000 steps, and the full radius walked at every decision (Tab. 7). On Large, radii 3 and 4 have the same mean SR across the two planners. On Giant a radius of 4 units reaches 0.98 with both planners against 0.86 and 0.82 at radius 3. Radius 4 therefore gives higher validation SR on Giant.

Table 7: Planning radius on the validation pairs (50 random pairs per maze, seed-0 state models, $\tau_{\max} = 8192$). SR with SPL in parentheses.

	Large		Giant	
Radius	PAP	GP	PAP	GP
2 units	0.88 (0.76)	0.94 (0.86)	0.76 (0.69)	0.76 (0.67)
3 units (used)	0.96 (0.84)	1.00 (0.89)	0.82 (0.73)	0.86 (0.74)
4 units	0.98 (0.89)	0.98 (0.90)	0.98 (0.85)	0.98 (0.82)

Horizon-Range Study. The horizon sweep of Fig. 2 reuses the trained representation and varies the largest horizon available during evaluation; it does not retrain a model for each maximum. Every point plans with every integer horizon from 1 to $\tau_{\max}$ on the five official tasks, with the same seeds as Tab. 1. This sweep is the one maze experiment that probes candidate outcomes in the simulator instead of the learned local model; Tab. 5 shows the two agree.

Additional egocentric Maze results. With horizons capped at 512, the 800-walk image-and-position model reaches the goal in every episode (SPL 0.77) and so does the position-only model (SPL 0.79); with every horizon to 8,192, the image model fails every episode and the position-only model succeeds in two. The 8M-chain models of Tab. 9 with the same cap: the position-only model succeeds in two episodes (SPL 0.56) and the image-and-position model fails all three, circling about 2 m from the goal; caps of 128 and 256 and a candidate radius of 1 m do not change this. The egocentric maze study compares images alone, images with position, images with heading, and images with both (Tab. 8).

Table 8: Egocentric Large-maze input ablation with GP, three training seeds.

		Pairs	
Input	Official SR	SR	SPL
image	0.13 ±0.04	0.43 ±0.02	0.18 ±0.06
+ position	0.35 ±0.32	0.52 ±0.34	0.37 ±0.37
+ heading	0.18 ±0.07	0.43 ±0.02	0.17 ±0.01
+ position + heading	**0.36** ±0.25	**0.62** ±0.27	**0.39** ±0.34

Table 9: Habitat corner tasks on the lattice, GP, every integer horizon 1 to 8,192, 5,000 steps; three training seeds (mean ± s.d.). The floor is a uniform choice among the same candidates.

Scene	States	Random floor	Position only	Image + position
Apartment 1	10,020	0.38	0.00 ±0.00	0.71 ±0.12

Additional Habitat results The result of Apartment 1's four corner tasks is on Table 9.

Van Gogh Room is too small for the corner tasks: the room is 3 m across, and the random floor already reaches 0.99 of the corner tasks within the 5,000-step budget, so its lattice numbers are not informative. Under the continuous protocol on the farthest reachable pair of its navmesh (4.15 m geodesic, three start headings, 200-step budget), image + position reaches SR 1.00 with SPL 0.90 and position only reaches SR 1.00 with SPL 1.00; the random floor also reaches every goal but with SPL 0.62.

D.4 Models, Data and Training setting

Model Architecture. Tables 10 and 11 specify the temporal encoders and training settings. A shared network conditions on a learned 32-dimensional horizon embedding through FiLM: $z_\tau = z \odot (1 + \gamma_\tau) + b_\tau$. The output head applies a linear layer, GELU, a second linear layer, and Softplus. Source embeddings are nonnegative and normalized to unit length. The main state and overhead-image models tie $g = h$; the first-person experiments also use a free, unnormalized goal

head. Position inputs are centered and scaled coordinate-wise. Optional position and heading inputs enter the image encoder through an additive linear projection; heading is represented by sine and cosine. The symmetric encoder runs learn a coefficient and a offset per horizon, initialized to 10 and -10, respectively.

Table 10: Temporal encoder architectures. All images are RGB.

Component	PointMaze state	PointMaze overhead	First-person / Habitat
Input	Position (x, y)	64×64 image	64×64 image with optional position/heading, or position plus heading $(x, y, \sin h, \cos h)$ alone
Backbone	MLP with GELU activatation	Four convolutions with linear projection	Four convolutions with linear projection
Horizon embedding	32 dimensions; one entry per integer horizon $1, \ldots, 8192$		

Training Objective and Optimization. All temporal encoders in Tab. 11 use Adam with learning rate 2×10^{-3}, coefficients $(0.9, 0.999)$, $\epsilon = 10^{-8}$, and no weight decay. Training lags follow a log-uniform distribution on $1, \ldots, 8192$.

Table 11: Temporal-model training settings. Updates count optimizer steps.

Experiment	Dataset / variation	Batch	Updates
State navigation	One 4M-step lattice walk per maze and seed	2048	240k
Overhead navigation	Same walk; 64×64 cell renderings	512	240k
Walk length	One walk of 0.1M, 0.4M, 1M, or 4M steps	2048	240k
State coverage	4M steps within a connected fraction $c \in \{0.25, 0.5, 0.75, 1\}$	2048	240k
Egocentric maze	800 walks of 20k steps; image, position, heading variants	512	40k
Habitat	One 8M-step walk per scene and seed, no stay, no reversal; position only, or image and position; free goal head	512	40k

Maze Data and Sampling. Each lattice-walk step chooses uniformly among eight compass moves and stay; a blocked move becomes stay. The lattice spacing is one world unit. World-model and offline-RL baselines instead use twenty continuous random-action episodes of 50k steps each. Data volume and collection dynamics therefore differ across method families. The overhead LeWM and DINO-WM runs use the same four rendered episodes (200k transitions) at 224×224 resolution.

For the walk-length study, we regenerate each trajectory to its specified length with the same random seed and walk rule. These walks share a prefix; they are not independent subsamples of transitions from the 4M-step dataset. The coverage study restricts the walk to a connected region grown from a seeded cell and replaces attempted exits by stay. It varies accessible states at a fixed 4M-step budget.

Egocentric and Habitat Data. The egocentric maze's five-action training walk samples stay, left turn, right turn, forward, and backward with probabilities $(0.10, 0.15, 0.15, 0.30, 0.30)$ on eight headings.

Habitat scenes are exported to a $0.25\,\mathrm{m}$ positional lattice with twelve headings; a turn changes the heading by 30°, and a forward or backward step moves two cells along the rounded heading direction, which gives twelve distinct moves at most 4° off the true heading. Van Gogh Room has 126 positions (1,512 states) and Apartment 1 has 835 (10,020 states). The training walk is a single 8M-step chain that never stays in place and never undoes its last move (no right turn after a left turn, no backward step after a forward one); it visits every state of both scenes (median 786 visits per state in Apartment 1). Models use the free goal head (Tab. 11) with either the image and the normalized position or the position and heading alone.

The continuous-protocol models of Fig. 3 and the main text instead train on 800 walks of 20,000 steps that also stay in place and step backward (action probabilities 0.10, 0.15, 0.15, 0.30, 0.30 for stay, left, right, forward, back). Figure 7 shows observations from the scene datasets.



Figure 7: Habitat training observations: four evenly spaced states of each scene's twelve-heading lattice.

Manipulation Model. The cube-single model keeps the source–target form and the NCE objective above but changes the encoder and how the horizon enters it (Tab. 12). The horizon embedding applies a two-layer MLP to sinusoidal features of $\log \tau$, as a diffusion model embeds its timestep, instead of a learned vector per integer horizon. The encoder is a ViT-Tiny-sized DiT that takes this embedding through adaLN in every block, so its CLS output depends on the horizon. A plain MLP projector follows, and its output is normalized to unit length. The per-horizon coefficient $\beta_0(\tau)$ and offset $\beta_1(\tau)$ then set the score's scale and level. Unlike the tied maze models, the source and goal branches share the encoder but use separate projectors, so $g \neq h$. Training horizons are drawn uniformly from $1, \ldots, 32$.

Table 12: Cube-single temporal model, expert-only condition.

Horizon embedding	Sinusoidal features of $\log \tau$ with MLP
Encoder	DiT with adaLN
Projector	MLP with LayerNorm; unit norm, either sign
Score	Learned $\beta_0(\tau)$ and $\beta_1(\tau)$
Training horizons	Uniform on $1, \ldots, 32$
Batch	512
Optimizer	AdamW, learning rate 3×10^{-4}, cosine schedule
Updates	30k on one GPU; 5.3 GPU-hours

Manipulation Data and Splits. Every cube-single training set holds 10,000 episodes of 201 frames, so the data volume is fixed and only its composition varies. The expert set is the one LeWM uses. The noisy set comes from the same scripted oracle with two perturbations. Temporally correlated Gaussian noise is added to the oracle's interpolated waypoint plan, with per-dimension scales $0.4 \times (0.05, 0.05, 0.05, 0.3, 1.0)$ over $(x, y, z, \text{yaw}, \text{gripper})$, smoothed in time with $\sigma = 0.5$. Each resulting action is then replaced with probability 0.1 by a uniform sample from the action space. Because the noise enters the plan rather than the action, the controller tracks a displaced path in closed loop instead of jittering around the correct one. The mixtures combine the two sets at expert fractions of 0, 25, 50, 75, and 100%.

D.5 Evaluation Protocols

Path Length and Controls. SPL $=$ (shortest path/actual path) $\times$ success. The shortest path is computed on a 0.25-unit grid with wall clearance and line-of-sight smoothing. The oracle follows the same geometry (clearance 0.8, step size 0.2). Random policy: sample a continuous action at each simulator step. Sampling uniformly from the planner's candidate set is a separate baseline.

Continuous Planning and Evaluation. Table 13 records the continuous PointMaze protocol, and Fig. 8 shows its twenty random start–goal pairs. Candidate endpoints come from a learned local dynamics model. Every candidate heading is rolled through the model for 15 steps, and the simulator is never queried before acting. Source and goal embeddings at off-grid positions use bilinear interpolation of the surrounding cell embeddings.

Table 13: Continuous PointMaze evaluation settings.

Setting	Official tasks	Random start–goal pairs
Episodes	Five tasks, 20 resets each	20 fixed pairs per maze, pair seed 5
Pair constraint	OGBench task definitions	Geodesic separation at least 20 cells
Action budget	1000 simulator steps	5000 simulator steps
Success	Agent enters the environment's one-unit goal disk	
Candidates	36 headings; radius 3 units	
Action block	15 learned-model steps, then 15 executed steps per decision	
Reachability	Keep candidates travelling at least 90% of the intended distance	
Motion filter	No stay; exclude headings within 45° of the reverse direction	
Metrics	SR and SPL	SR and SPL

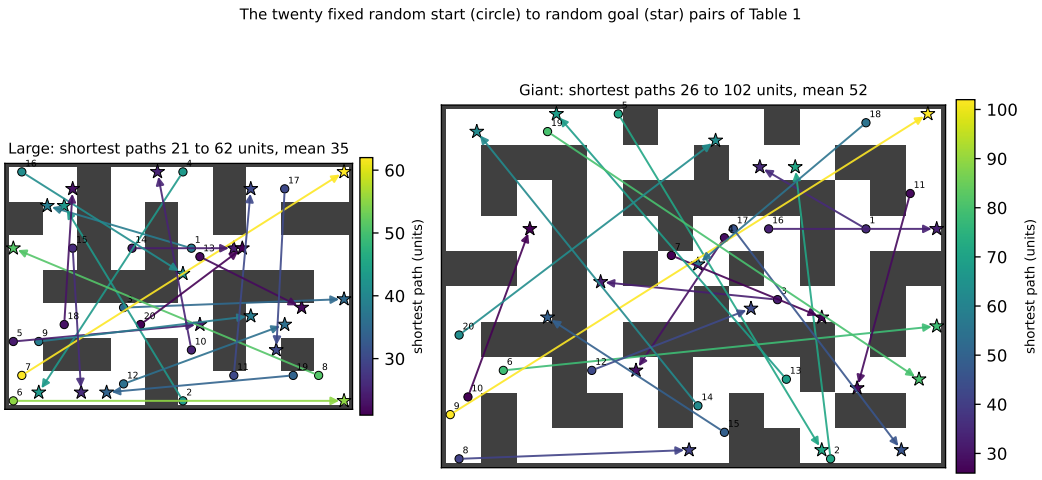

Figure 8: The twenty start (circle) to goal (star) pairs of Tab. 1 on each maze, coloured by shortest-path length; numbers label the pairs. Walls are dark.

Egocentric Inputs and Habitat. In Habitat, the four corner tasks connect the two most distant reachable corner pairs in both directions; each task runs 25 episodes from a random heading within 0.5 m of the corner, with a 5,000-step budget, the planner of Tab. 13 restricted to the five lattice actions with the previous state excluded, and success within one lattice move of the goal. The random floor chooses uniformly among the same candidates.

The continuous Habitat protocol of Fig. 3 plans over 36 headings: candidates lie 0.75 m away, are probed on the scene's navmesh recomputed for an agent radius of 0.2 m (a candidate is kept if the agent travels at least 90% of the way), the chosen heading is walked to its candidate (three 0.25 m steps) before re-planning, with no constraint on consecutive headings, the goal is a 0.3 m disk, and SPL uses the navmesh geodesic. Frames are rendered at every candidate. The pair runs from the lower right of the apartment to its upper right (6.5 m apart, three episodes with start headings 0°, 120° and 240°, budget 3,000 steps).

Manipulation. Cube-single (Park et al., 2025) places one cube on a table in front of a 5-DoF arm with actions in $[-1, 1]^5$. Observations are 224×224 RGB renders from a fixed front camera. The goal is the observation 25 steps later in the same trajectory, and the budget is 50 steps, as LeWM specifies for cube. An episode succeeds when the cube center comes within 0.04 m of its goal position,

checked at every step. Table 3 reports two settings that should not be compared with each other. The mixed-data block draws 1,000 episodes from the held-out split (Sec. D.4); LeWM is trained on each mixture with its released code, and spreads are standard deviations over ten runs with different seeds. The expert-only block follows the published baselines; our model is evaluated there with LeWM's evaluation code. Under our held-out setting, LeWM's released checkpoint reaches 0.71, against the 0.74 reported by Maes et al. (2026). At every step, the planner samples 512 actions uniformly from $[-1, 1]^5$ and holds each for 5 steps, matching LeWM's action block. Then it scores each resulting observation with Eq. (8) over every integer horizon from 1 to 32, executes the best action for one step, and replans.

Baseline Settings. Table 14 groups the comparisons by model family. Published NAVIGATE results come from Park et al. (2025); random-data GCIQL, QRL, HIQL, and HILP are our runs. Baselines retain their own action-selection procedures.

Table 14: Baseline configurations.

Family	Method	Training and action selection
Offline RL	GCIQL, QRL, HIQL	OGBench reference implementations; 10^6 gradient steps, batch 1,024; deterministic actions
	HILP	Authors' code on OGBench data; 10^6 gradient steps, batch 1,024, 32-dimensional skills; deterministic actions
World model	LeWM (state)	20 epochs; frameskip 5, history 3, one-step prediction; CEM
	LeWM (vision)	3 epochs; learned visual encoder; CEM with 300 samples, 30 elites, 10 iterations
	DINO-WM	3 epochs; frozen DINOv2 features; CEM with 100 samples, 5 iterations
	DINO-WM + position	Same model; planning cost adds the position term with weight 1
Control	Random policy	Uniform continuous actions at every simulator step
	Shortest-path oracle	Continuous geometric path to the goal disk

All offline RL baselines train on the same random-action data as the world models: twenty episodes of 50,000 steps per maze, one million transitions. Each method is trained with three seeds, and tables report the mean over seeds. QRL uses the PointMaze hyperparameters of Park et al. (2025), with $\alpha = 0.0003$ and discount 0.995 on Giant. GCIQL and HIQL use the default configurations of the reference implementations. HILP uses the hyperparameters of its released code. At test time each policy acts on the current position and the goal position, with the same budgets and success test as our model.

Vision LeWM and DINO-WM plan five blocks ahead, with five simulator steps per block, and replan after each block. DINO-WM receives the point's (x, y) position as its proprioceptive input alongside each frame, as in its released PointMaze configuration. Its planning cost is the distance between predicted and goal visual features plus α times the distance between predicted and goal proprioceptive embeddings. The released configuration sets $\alpha = 0$; the DINO-WM + position row uses $\alpha = 1$ with the goal position given, and is otherwise identical. DINO-WM uses 100-step training chunks; LeWM receives the corresponding frames. The reported state LeWM best run uses 1000 CEM samples, 30 iterations, a ten-block horizon, and 20 evaluation episodes. Its evaluation count differs from our 100-episode official-task evaluation.

For all manipulation baseline models on LeWM's evaluation protocol, we adapted the results from Maes et al. (2026).

Reproducibility on different seed Every number of our models and of the reproduced baselines in Tab. 1 is a mean over training seeds, with the official tasks and the random pairs evaluated on the same seeds. Table 15 lists the success of each seed on the official tasks. The main results in Tab. 1 use every integer horizon from 1 to 8192.

Table 15: Official-task SR of every training seed with every integer horizon from 1 to 8192 and the learned local dynamics, for the state and the image models of Tab. 1.

Maze	Planner	Seed 0	Seed 1	Seed 2
Large	PAP ($\beta \to \infty$)	0.75	0.61	0.92
Large	GP ($\beta \to 0^+$)	0.96	0.80	1.00
Giant	PAP ($\beta \to \infty$)	0.88	0.90	0.60
Giant	GP ($\beta \to 0^+$)	0.76	0.86	0.67
Large, images	PAP ($\beta \to \infty$)	0.58	1.00	—
Large, images	GP ($\beta \to 0^+$)	0.84	0.97	—
Giant, images	PAP ($\beta \to \infty$)	0.36	0.49	0.73
Giant, images	GP ($\beta \to 0^+$)	0.37	0.24	0.57