LEARNING COMMUTE-TIME-PRESERVING WORLD MODELS FOR PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

World models allow agents to plan in latent space by choosing a sequence of actions that most reduces the distance to a given goal state. Thus, planning can benefit from latent representations whose distances mirror commute-times in the environment. The spectral embedding space of the graph Laplacian provides such a representation, if it obeys a specific eigenvalue-dependent scaling. Unfortunately, instantiating the graph Laplacian is intractable in large, continuous environments. Self-supervised learning offers a natural route to such commute-time-preserving embeddings at scale. However, here we show that existing methods, which commonly encourage isotropic representations to prevent representational collapse, tend to degrade the "correct" eigenvalue-dependent scaling, leading to an inaccurate representation of commute times. To address this problem, we introduce Commute-Time-Preserving World Models (CTWMs), combining a latent displacement predictor and a log-determinant regularizer that prevents collapse, which provably recover the correctly scaled Laplacian representation under deterministic dynamics and at the predictor's fixed point. In numerical simulations, CTWM matches or outperforms LeWM, a task-agnostic baseline, on several complex, continuous goal-reaching benchmarks, while using half the parameters.

1 INTRODUCTION

Acting intelligently in complex environments requires imagining the future. An agent that can simulate the consequences of its actions before taking them can plan, rather than merely react. This is the promise of model-based reinforcement learning (RL), where a learned world model might enable fast RL and generalization (Sutton, 1990; Deisenroth & Rasmussen, 2011; Chua et al., 2018; Janner et al., 2019). Still, the world models behind seminal successes (e.g., Schrittwieser et al., 2020) are shaped end-to-end by a single, fixed reward function, and adapt poorly when the objective changes (van Seijen et al., 2020). Animals do not share this inflexibility. For instance, a rat allowed to explore a maze in the absence of any reward will act as it possesses a full map of the maze the instant a reward is introduced (Tolman & Honzik, 1930; Tolman, 1948). Such *latent learning* suggests that nature has settled on task-agnostic world models as a strategy. What does it take for artificial agents to do the same, reliably and at scale?

Ha & Schmidhuber (2018) took the first step by learning representation and dynamics separately by first training a variational autoencoder first and then fitting a recurrent dynamics model to it afterward. Dreamer and its successors instead train both jointly, but through reconstruction (Hafner et al., 2020; 2022; 2025), which can bias representations on uninformative nuisance variables required to redraw the scene pixel-by-pixel (Deng et al., 2021). More recently, Maes et al. (2026b) introduced Le World Model (LeWM), a task-agnostic world model trained jointly, end-to-end, *without* reconstruction, using only a lightweight regularizer to prevent collapse. This work provides a strong baseline for learning task-agnostic representations, but collapse-avoidance alone does not naturally lead to representational geometries that are also useful for planning (Wang et al., 2026; Nguyen et al., 2026).

What properties should a representation suited for planning have? In classical RL on discrete state spaces there exists a principled choice of representation geometry: the *Laplacian representation*, which embeds states using eigenvectors of the transition graph's Laplacian (Mahadevan & Maggioni, 2007), and under an appropriate eigenvalue-dependent rescaling, recovers the graph's

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.

commute-time distance in its Euclidean distances (Xiao & Gutman, 2003; Qiu & Hancock, 2007). This distance accounts for temporal distances between states, and thus forms a natural basis for planning, which has to take obstacles and other relevant factors of the environment into account. Learning this representation directly from data is well understood (Wu et al., 2018; Wang et al., 2021; Gomez et al., 2024), with applications to option discovery (Machado et al., 2017) and planning (Wang et al., 2022).

What is missing is a bridge between these two lines of work: the question is whether the same reconstruction-free, joint-embedding recipe that makes LeWM scale can also be steered toward a representational geometry that is optimal for planning, a question that has not, to our knowledge, been investigated. We show that this is possible and introduce Commute-Time-Preserving World Model (CTWM). Our key contributions are:

- We introduce CTWM, a reconstruction-free world model that jointly learns an encoder and a latent dynamics model by predicting action-induced latent displacements and applying an entropy-maximizing regularizer.
- We prove that, under deterministic and reversible dynamics, CTWM converges exactly to the Laplacian representation of the environment's transition graph, correctly scaled such that latent distances equal commute-time distances.
- On discrete environments with a known transition graph, we confirm this directly: the learned embeddings' pairwise distances match ground-truth commute times, whereas methods encouraging isotropic representations, e.g., SIGReg, do not recover the correct scaling.
- Finally, we show that CTWM outperforms LeWM on half the parameter budget across LeWM's own goal-reaching benchmarks and two additional visual benchmarks from OGBench (Scene and Pointmaze), all evaluated from pixel observations with no oracle goal representation, which puts the full burden of representation learning on the encoder.

2 Related Work

World models support sequential decision-making by letting an agent maintain a belief over hidden states, improve a policy through imagined rollouts rather than environment interaction, and plan directly by simulating candidate action sequences at test time.

World models for policy learning. Ha & Schmidhuber (2018) trained a VAE as an encoder and then a predictive model on its latent space to train an agent entirely in imagination. More recently, the Dreamer family brought this idea to scale allowing policy iteration in complex environments (Hafner et al., 2020; 2022; 2025). While the above works relied on reconstruction objectives for representation learning, the notion was further extended to the reconstruction-free setting (Burchi & Timofte, 2024; Hauri & Zenke, 2026).

World models for planning. To plan directly by simulating action sequences, classic work searched decision trees of future moves using known environment dynamics (Silver et al., 2016), later extended to learned transition models when the true dynamics are unavailable (Schrittwieser et al., 2020). However, these approaches depend on reward and thus on the policy. This dependence typically precludes zero-shot generalization, as it requires retraining the model when the task or the rules of the game change.

Task agnostic world models address this limitation by learning representations and transition models from data without requiring reward. One line of work first trains an encoder network on generic image or video data (Zhou et al., 2024; Assran et al., 2025) using reconstruction-free self-supervised learning (SSL) and then learns a latent dynamics model on top for planning. More recently several studies relied on joint-embedding predictive architectures (JEPAs), allowing the encoder and predictor, i.e., the dynamics model, to be trained jointly (Sobal et al., 2026; Maes et al., 2026b; Zhao et al., 2026), thereby further boosting planning performance and enabling zero-shot generalization. To mitigate the compounding rollout error such joint-embedding predictors accumulate over long horizons, some models predict jumpy, temporally abstract transitions instead (Farebrother et al., 2025; 2026). The main difference of the above reconstruction-free approaches is how they avoid collapse.

Avoiding collapse in SSL: To prevent trivial solutions in SSL different methods use distinct mechanisms. On the one hand, contrastive methods pull together the representations of positive sam-

ples while pushing apart negative ones (Sohn, 2016; Oord et al., 2018; Chen et al., 2020). On the other hand, non-contrastive methods forgo negatives and either rely on asymmetric architectures using a predictor and a stop-gradient, for instance, SimSiam (Chen & He, 2021) and BYOL (Grill et al., 2020), or additional regularization (Barlow Twins; Zbontar et al., 2021) as well as variance-invariance-covariance regularization (VICReg) (Bardes et al., 2021). More recently Balestriero & LeCun (2025) proposed Sketched Isotropic Gaussian Regularization (SIGReg) that matches the full projected embedding distribution to an isotropic Gaussian. Most existing SSL methods encourage isotropic representations either explicitly or implicitly (Halvagal et al., 2023), which can pose challenges for learning scaled representations.

Laplacian representations embed states using eigenvectors of the graph Laplacian induced by the Markov Decision Process (MDP)'s transition structure (Mahadevan & Maggioni, 2007), and are closely related to slow feature analysis (Wiskott & Sejnowski, 2002; Sprekeler, 2011; Richthofer & Wiskott, 2015). In RL, Wu et al. (2018) proposed an objective for learning these representations directly from data, which was improved subsequently (Wang et al., 2021). Gomez et al. (2024) introduced augmented Lagrangian Laplacian objective (ALLO), a method explicitly designed to recover both the Laplacian eigenvectors and eigenvalues from sampled transitions through a constrained optimization scheme. Laplacian representations have served to define spatio-temporally abstract states (Shehmar et al., 2026) and intrinsic reward functions (Machado et al., 2017). Connections between SSL objectives and spectral embedding methods were already studied by Balestriero & LeCun (2022). Although the above methods recover the Laplacian geometry, they neither necessarily learn a latent dynamics model nor guarantee appropriate eigenvalue-dependent scaling.

Alternative latent geometries for goal-reaching. A separate line of work shapes latent representations directly around planning-relevant distances, by learning (quasi-)metric embeddings via temporal-difference estimation of the transit time between states and goals (Wang et al., 2023; Myers et al., 2026; Park et al., 2026; Zheng et al., 2025). Unlike our method, these models only compute temporal distances between states, but do not train a world model with latent state predictor end-to-end. A second body of work learns successor features that aggregate discounted future state occupancies (Eysenbach et al., 2022; Bagatella et al., 2026). While these methods enable zero-shot goal-reaching, their learned distances are inherently policy-dependent rather than the intrinsic structure of the environment. In contrast, CTWM recovers the environment's commute-time distance.

3 Constructing Commute-Time-Preserving World Models

Before we start, we briefly introduce the basic notation. Let $\mathcal{A}$ be an action space and $\mathcal{S}$ an observation space. An encoder $\phi : \mathcal{S} \to \mathcal{Z}$ maps observations to a latent space $\mathcal{Z}$, and a predictor $T : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$ simulates the dynamics in that space, $z_{t+1} = T(z_t, a_t)$.

3.1 Planning in metric latent spaces

Planning refers to finding the action sequence that minimizes a cost J over a horizon H. This cost can be reward-based, as in Hansen et al. (2024), or reward-free, as is common in goal-conditioned reinforcement learning. In the latter case, the cost measures the L_2 distance between the final latent state and an encoded goal $z_g = \phi(g)$:

$$a^\star_{1:H} = \arg\min_{a_{1:H}} J(z_H, z_g) = \arg\min_{a_{1:H}} \|z_H - z_g\|_2 \quad , \tag{1}$$

where z_H is obtained by recursively applying the learned dynamics $z_{t+1} = T(z_t, a_t)$ from $z_0 = \phi(s_0)$. This problem is commonly solved with the sampling-based cross-entropy method (CEM) (Rubinstein & Kroese, 2004), or gradient-based methods (V et al., 2023). Common to both approaches is that they rely on computing distances in the latent space, which requires these distances to be meaningful in the environment. A prominent choice is to require points in the latent space to preserve commute-time distances in the environment (Xiao & Gutman, 2003; Qiu & Hancock, 2007). On a graph, the commute time $C(s, s')$ corresponds to the expected number of steps for a random walk to travel from node s to node s' and back. It relates to the eigenvalues and eigenvectors $(\lambda_k, u_k)_{k\geq 1}$ of the graph Laplacian through

$$C(s, s') = \mathrm{vol}(G) \sum_{k\geq 1} \frac{\big(u_k(s) - u_k(s')\big)^2}{\lambda_k} \quad , \tag{2}$$

where $\mathrm{vol}(G)$, the volume of the graph, is the sum of its node degrees (Qiu & Hancock, 2007). For latent distances to correspond to commute time distance, we further define the scaled Laplacian representation (Qiu & Hancock, 2007; Wang et al., 2022; Shehmar et al., 2026) as $\psi_k(s) := \lambda_k^{-1/2} u_k(s)$, which yields

$$C(s, s') = \mathrm{vol}(G)\, \|\psi(s) - \psi(s')\|_2^2 \quad , \tag{3}$$

so that squared L_2 distances in ψ are commute-time distances up to a global multiplicative constant.

In other words, there exists a representation space in which the commute-time distance is Euclidean. It requires an eigenvalue-dependent scaling of the representations, which is realized by the scaled Laplacian representation. This property makes the scaled Laplacian representation well-suited for planning and thus a desirable learning target for SSL algorithms whose embeddings should facilitate planning in downstream tasks. Crucially, the scaled Laplacian representation is anisotropic for most environments with non-degenerate eigenvalues. In contrast, existing SSL methods commonly encourage isotropic embeddings. Before we can study whether and to what extent different SSL methods recover the scaled Laplacian representation, we first present the general framework for learning task-agnostic world model with SSL.

3.2 Learning task-agnostic world models with SSL

Given a task-agnostic dataset of transitions $\{(s_k, a_k, s'_k)\}_{k=1}^N$, collected without rewards or task labels by an arbitrary behavior policy, the canonical approach to learning a world model via SSL is specified by the tuple (ϕ, T), in which the encoder ϕ maps a state to a latent representation $z = \phi(s)$, and the predictor T estimates the effect of an action in latent space, $z' = T(z, a)$. Both components are typically learned by minimizing

$$\mathcal{L}(\phi, T) = \mathbb{E}_{(s,a,s')} \left[\|T(\phi(s), a) - \phi(s')\|^2 \right] + \mathrm{Reg}(\phi) \quad , \tag{4}$$

where $\mathrm{Reg}(\phi)$ denotes a regularizer that prevents representational collapse. For instance, LeWM (Maes et al., 2026b) relies on SIGReg (Balestriero & LeCun, 2025), but other regularizers are possible. Importantly, the regularizer may also be realized implicitly, e.g., through a stop-gradient operation (Grill et al., 2020; Tian et al., 2021; Halvagal et al., 2023), as long as it preserves representational entropy (Shwartz-Ziv et al., 2023; Mikulasch & Zenke, 2026).

Our CTWM approach deviates from the formulation above in two distinct ways: We use residual prediction and a specific log-determinant (log-det) regularization term to prevent collapse, which we will explain in the following two sections.

3.3 Learning world models via residual predictions

The first difference of CTWM to existing approaches such as LeWM is residual prediction. Concretely, the predictor T is trained to predict the residual or displacement between successive embeddings $\Delta(s, s') = \phi(s') - \phi(s)$ such that the next embedding is given by $\phi(s') = \phi(s) + T(\phi(s), a)$ (Figure 1a). The encoder and displacement are then trained jointly by minimizing:

$$\mathcal{L}(\phi, T) = \mathbb{E}_{(s,a,s')}[\|\Delta(s, s') - T(\phi(s), a)\|^2 + \beta \|T(\phi(s), a)\|^2] + Reg(\phi) \quad , \tag{5}$$

where the $\beta \|T(\phi(s), a)\|^2$ term acts as a regularizer that penalizes large changes (Saanum et al., 2024), which can also be seen as a slowness objective (Wiskott & Sejnowski, 2002; Sprekeler, 2011). Without this term, a sufficiently expressive predictor can absorb arbitrarily large latent displacements, leaving the encoder free to map temporally consecutive states to distant points in the latent space. By penalizing large predictor outputs, $\beta > 0$ forces the encoder to keep consecutive latent states close together: the only way to simultaneously achieve low prediction error and low predictor norm is to produce small displacements. Where required, the above loss can be equivalently written as a next-state prediction problem instead of residual prediction, by simply adapting the regularizer (see Appendix D.1).

For deterministic dynamics, one can show that the embeddings z_t resulting from optimizing equation 5 converge to the Laplacian subspace if the regularization $Reg(\phi)$ introduces an orthonormality constraint (Appendix C.1). To understand this intuitively, it is helpful to assume that the predictor is

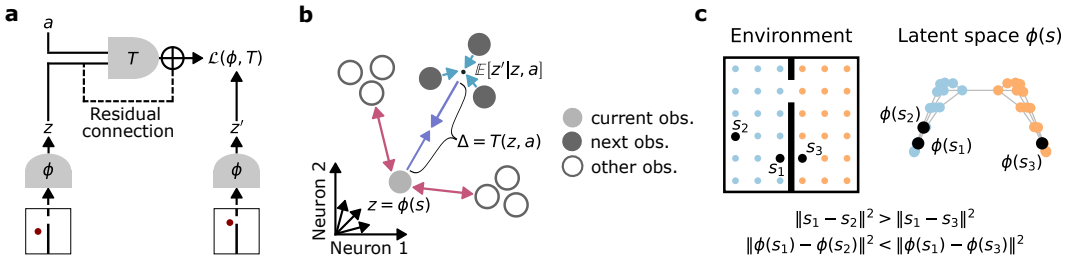


Figure 1: **(a)** Overview of CTWM: a shared encoder ϕ maps observations s and s' to the representations z and z' respectively. The transition model T conditioned on action a, predicts a residual displacement Δ that is added to the current embedding via a residual connection. The loss $\mathcal{L}(\phi, T)$ is computed from the predicted and actual next-state embeddings. **(b)** Geometric interpretation of the loss $\mathcal{L}(\phi, T^\star)$ in the latent space; arrow colors refer to terms in equation 6. The loss terms encourage predictable transitions (teal), penalize large distances between consecutive points (blue), and prevent collapse of representation (magenta). **(c)** Latent distances reflect commute-time distance rather than spatial proximity. Left: Two-room connected by a door; s_1 and s_3 are adjacent but separated by a wall (dashed line). Right: in the CTWM embedding the rooms unfold around the door, placing s_1 near s_2 and far from s_3.

at its fixed point of learning $T^\star = \frac{\mathbb{E}[\Delta|s,a]}{1+\beta}$. This assumption allows us rewrite the loss as follows

$$\mathcal{L}(\phi, T^\star) = \mathbb{E}_{(s,a)}[\mathrm{tr}\big(\mathrm{Cov}(\Delta \mid s, a)\big)] + \frac{\beta}{1+\beta}\mathbb{E}_{(s,a)}[\|\mu(s,a)\|^2] + Reg(\phi, s) \quad , \tag{6}$$

where $\mu(s, a) = \mathbb{E}[\Delta|s, a]$ is the mean conditioned displacement and the colors correspond to the diagram in Figure 1b. Given a stochastic transition function $p(s'|s, a)$, the first term tries to make $p(z'|z, a)$ predictable by making the sum of variances small, all the while the second term attracts states that are correlated in time together. Finally, the last term prevents global collapse. The fixed point condition also implies that we should use $(1 + \beta)T$ for rollouts instead of T (see Appendix D.2).

3.4 Regularization to prevent collapse

The second destinction of CTWM from previous work is that it uses log-det as a regularizer $Reg(\phi)$ to prevent collapse. The underlying reasons is that different regularizers converge to different representations. For example, a hard orthonormality constraint (Appendix B.1) yields a Laplacian representation, but without eigenvalue-dependent scaling by design (Wu et al., 2018, Appendix C.2). The constraint can be relaxed through VCReg (Zhu et al., 2023), which enforces a minimal floor on the diagonal of the covariance matrix (Appendix B.2). Assuming that this regularizer converges to the Laplacian representation, the eigenvalue-dependent scaling is constant for the smallest eigenvalues and $\propto \lambda_k^{-1}$ for the rest (Appendix C.2). Neither scaling corresponds to the scaled Laplacian representations and, thus, does not represent commute-times accurately.

To address this issue, we use log-det regularization. Crucially, it does not impose a hard threshold on feature variance and VCReg constitutes an approximation of this quantity when the representational covariance matrix is close to diagonal (Shwartz-Ziv et al., 2023; Mikulasch & Zenke, 2026). We will show in the next section that log-det gives representations enough flexibility to achieve $\lambda_k^{-1/2}$ scaling, as required for capturing commute-time distances.

3.5 Convergence to the scaled Laplacian representation

Combining residual prediction with log-det regularization yields the CTWM training objective:

$$\boxed{\mathcal{L}(\phi, T) = \mathbb{E}_{(s,a,s')}\left[\|\Delta(s, s') - T(\phi(s), a)\|^2 + \beta\|T(\phi(s), a)\|^2\right] - \frac{\gamma}{2}\log|\Sigma|\,,} \tag{7}$$

with the two hyperparameters β and γ and the covariance matrix $\Sigma \equiv \mathrm{Cov}\big(\phi(s)\big)$. In the following we will show that this objective is well suited for learning world models for planning. Our main result is that, at the optimum, the encoder recovers the scaled Laplacian representation.

Theorem 1 (Recovery of scaled Laplacian representation). *Let $(\lambda_k, u_k)_{k\geq 0}$ be the eigenvalues and eigenvectors of the Laplacian of the transition graph, ordered as $0 = \lambda_0 < \lambda_1 \leq \cdots \leq \lambda_{n-1}$ with $\lambda_d < \lambda_{d+1}$, and let*

$$\psi(s) = \left(\lambda_k^{-1/2} u_k(s)\right)_{k=1}^{d}, \qquad d = \dim \mathcal{Z}, \tag{8}$$

be the scaled Laplacian representation. Under deterministic dynamics, and assuming the predictor is at its fixed point $T^\star = \frac{\mathbb{E}[\Delta|s,a]}{1+\beta}$, equation 7 becomes:

$$\mathcal{L}(\phi, T^\star) = \frac{\beta}{1+\beta}\, \mathbb{E}_{(s,s')}\left[\|\phi(s') - \phi(s)\|^2\right] - \frac{\gamma}{2}\log|\Sigma|, \qquad \Sigma \equiv \mathrm{Cov}\left(\phi(s)\right), \tag{9}$$

and is minimized, up to a rotation and a translation, by

$$\phi^\star(s) = \sqrt{\tfrac{\gamma(1+\beta)}{4\beta}}\, \psi(s) \quad . \tag{10}$$

The proof is given in Appendix C.3. The eigendirections are the slowest ones of the Laplacian, so they are shared with VCReg (Balestriero & LeCun, 2022) and orthonormality constraints (Wu et al., 2018), but what distinguishes the log-det from the two other regularizers is the eigenvalue-dependent scaling $\lambda_k^{-1/2}$, which carries the metric information relevant for planning. This relationship between the commute-time distance to the L_2 distance in the learned space can be made explicit as follows:

Corollary 1.1 (Relationship to commute-time distance). *Let*

$$C_d(s, s') := \sum_{k=1}^{d} \frac{\left(u_k(s) - u_k(s')\right)^2}{\lambda_k} \tag{11}$$

be the rank-d truncation of the commute-time distance on the transition graph (Shehmar et al., 2026), then

$$\left\|\phi^\star(s) - \phi^\star(s')\right\|_2^2 = \frac{\gamma(1+\beta)}{4\beta}\, C_d(s, s') \qquad \textit{for all } s, s' \in \mathcal{S}. \tag{12}$$

This result follows trivially from combining equation 10 with equation 11. To verify this relation experimentally, we further link the eigenvalues of the underlying graph to the variance of the representations:

Corollary 1.2 (Variance of the representation). *The k-th eigendirection of $\phi^\star$ carries variance*

$$\mathrm{Var}\left(\phi_k^\star(s)\right) = \frac{\gamma(1+\beta)}{4\beta\, \lambda_k} \quad . \tag{13}$$

In other words, slow directions are amplified and fast ones inhibited (see Appendix C.4 for the proof). For comparison we compute the ratios $\mathrm{Var}(\phi_k(s))/\,\mathrm{Var}(\phi_1(s))$, which are equal to λ_1/λ_k for our loss, to get rid of the global scale factor. Together, these results show that regularizers for preventing representational collapse may result in differently scaled representations. Specifically, the analysis shows that orthonormality constraints and VCReg are not consistent with the scaled Laplacian representation, whereas the log-det supports a correctly scaled representation, with variance $\propto \lambda_k^{-1}$ and L_2 distances equal to commute times (Figure 1c). In the following, we will confirm these findings in numerical experiments.

4 Experiments

We first sought to verify that CTWM learns successfully in a setting in which the the ground-truth Laplacian representation can be computed explicitly. To this end, we constructed a simple environment in which the agent moves along the edges of a discrete toroidal graph (Fig. 2a). Each node on the graph corresponded to an image displaying a unique three digit sequence, which were randomly assigned to the nodes. To make this a non-trivial representation learning problem, the digits' images were drawn from the MNIST dataset (LeCun et al., 1998) and every time the agent revisited a given state, a different digit corresponding to the same class were drawn. The underlying graph dynamics are thus deterministic, while the observations, and hence the transitions in observation space, are stochastic, requiring the encoder to learn digit representations that generalize.

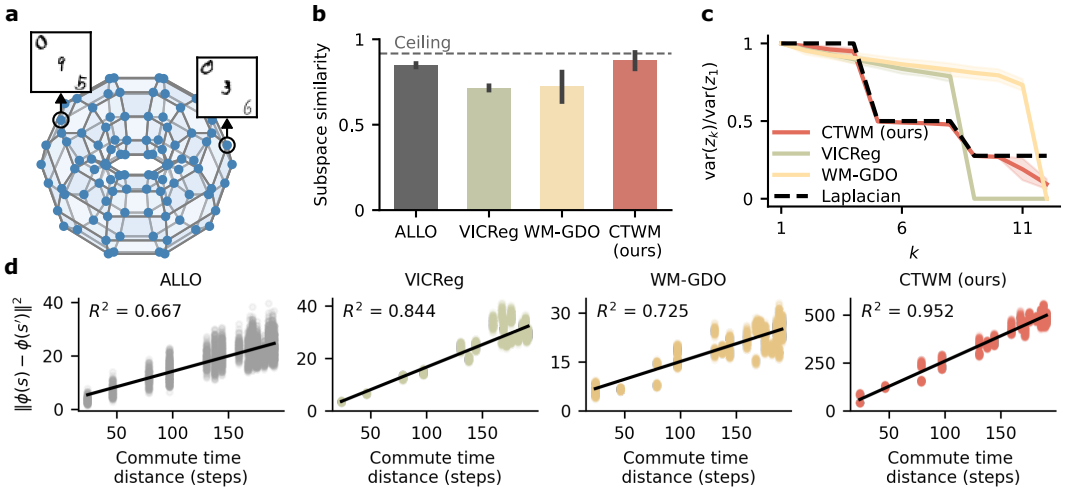


Figure 2: **Torus environment and empirical verification of the theoretical predictions.** **(a)** The torus graph: each vertex carries a unique three-digit label, randomly shuffled across the graph, observed as three MNIST images sampled independently from the corresponding classes at each visit. **(b)** Subspace similarity to the ground-truth Laplacian eigenbasis for the four methods $\pm$ SD ($n = 4$; see Appendix F); the dashed line indicates the ceiling imposed by the degenerate eigenspectrum. **(c)** Variance ratio $\mathrm{var}(z_k)/\mathrm{var}(z_1)$ against eigenvalue index k (cf. Corollary 1.2), with the ground-truth scaling λ_1/λ_k in dashed black. Only CTWM tracks it, confirming that the log-det regularizer recovers the $\lambda_k^{-1/2}$ scaling; VICReg and WM-GDO show more isotropic spectra, as encouraged by their respective regularizers. **(d)** Squared L_2 latent distance versus ground-truth commute-time distance, one point per pair of graph nodes. Black line: least-squares linear fit, with its coefficient of determination R^2. CTWM achieves the highest R^2.

We trained a small CTWM model on 20,000 transitions in the above environment using a random policy. The encoder was a small convolutional network that produced a 12-dimensional embedding, and the predictor was a single-hidden-layer MLP with 32 units. The action was one-hot encoded and we set the displacement penalty to $\beta = 0.1$. All simulations were run for $n = 4$ seeds.

To check whether CTWM had learned the correct representational subspace shared with the actual spectral embedding of the graph Laplacian and the correct eigenvalue scaling, we computed the subspace similarity of the learned embeddings to the analytical solution. For comparison, we further computed the subspace similarity for VICReg, as a representative example of non-residual world-modeling approaches, and ALLO (Gomez et al., 2024), which served as a proxy of the Laplacian representation. We found that CTWM's subspace similarity was notably higher than VICReg and on par with ALLO (Fig. 2b). We wondered whether this difference could be explained by residual prediction alone. To address this question, we trained a model, called WM-GDO, that keeps CTWM's residual prediction while replacing its log-det regularization with a graph-drawing orthonormality constraint (Koren, 2003; Wu et al., 2018). This modification resulted in comparable subspace similarity to VICReg. Thus residual prediction alone is not sufficient for good subspace alignment.

Moreover, neither ALLO nor CTWM reach a subspace similarity of one. We reasoned that his is a consequence of a degenerate eigenvalue Laplacian spectrum on the torus: every non-trivial eigenvalue has a multiplicity ≥ 4 and, hence, the eigenbasis is not uniquely determined. This degeneracy puts an inherent ceiling on the subspace similarity metric at ≈ 0.91, estimated by randomly rotating the eigenvectors within each degenerate block. CTWM and ALLO approach this ceiling.

We next quantified the relative variance explained per principal component for the different approaches. This analysis revealed that CTWM's spectrum closely followed the ground-truth Laplacian (Fig. 2c). In contrast, VICReg and WM-GDO displayed markedly different eigenvalue scaling. Finally, we checked how the resulting representation differences affect the encoding of commute time distances. To that end, we compared ground-truth commute times with the corresponding embedding distances for all algorithms. CTWM represented commute times most faithfully ($R^2 = 0.95$; Fig. 2d), followed by VICReg ($R^2 = 0.84$), WM-GDO ($R^2 = 0.73$), and ALLO

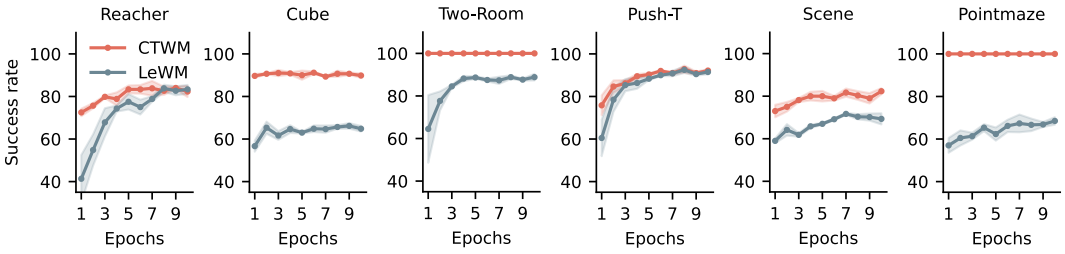


Figure 3: **Success rates of CTWM and LeWM on the six benchmark tasks:** Reacher, Cube, Two-Room, PushT, Scene and Pointmaze. CTWM (red) matches or outperforms LeWM (blue) on every environment using half the number of parameters. Shaded regions indicate the SD ($n = 3$).

($R^2 = 0.67$). Thus, CTWM learns the Laplacian spectral space, the correct eigenvalue scaling, and its embeddings faithfully encode commute-time distances.

4.1 Learning CTWM in continuous RL environments

To evaluate whether the advantage of representing commute-time geometry transfers beyond discrete graphs, we compared CTWM against LeWM (Maes et al., 2026b) on the following six benchmarks tasks. For (i–iv) we used the dataset provided by Maes et al. (2026a), while for (v, vi) we used two environments from (Park et al., 2025). All environments provide pixel observations. (i) *Two-Room*: 2D navigation between two rooms joined by a door (Sobal et al., 2026), with trajectories from a noisy heuristic policy. (ii) *Reacher*: a two-link arm reaching a target point in the plane (Tassa et al., 2018), with trajectories from a trained soft actor-critic policy. (iii) *PushT*: a point agent pushing a T-shaped object to a target pose (Zhou et al., 2024), with expert demonstrations. (iv) *Cube*: 3D manipulation in which a robot arm moves a cube between positions, with trajectories from the heuristic policy of the benchmark (Park et al., 2025). (v) *Pointmaze*: 2D maze navigation, using the 1000 trajectories of the OGBench medium-navigate dataset (Park et al., 2025) converted to visual inputs (Figure 4a). (vi) *Scene*: 3D manipulation with diverse objects (drawer, window, cube, buttons) (Park et al., 2025), with data from Maes et al. (2026a).

As in Maes et al. (2026b), we predicted 25 steps ahead with a maximum planning budget of 50 steps. To implement CTWM, we replaced LeWM's SIGReg training objective with equation 7, while using the same ViT-T encoder with 5.5M parameters. For the predictor, we retained the deep-AdaLN architecture with six layers but reduced the latent dimension from 192 to 64, which decreased the predictor's parameter count from 11M to 3M. The total model size was 9M parameters, compared to 18M for LeWM. As T is already normalized, we further removed the weight decay for CTWM. Instead of batch normalization, required for SIGReg, we used layer normalization (Ba et al., 2016).

4.1.1 CTWM results in superior planning performance

Following the steps of Maes et al. (2026b), we evaluated planning accuracy using CEM on models trained with LeWM and CTWM on all six environments for $n = 3$ seeds each, using the same hyperparameters across all of them. CTWM performed either better than, or on par with LeWM on all tasks (Fig. 3; Table 3). On Two-Room, where Maes et al. (2026b) noted that SIGReg can be detrimental in low-complexity environments, CTWM reached 100% success rate, matching the ceiling obtained by other methods. On Reacher and PushT, both methods achieved comparable performance. The clearest gains appeared on the three harder tasks: Cube, Scene, and Pointmaze, on which CTWM consistently outperformed LeWM, with larger margins during early training.

Given the high success rate on several benchmarks, we repeated planning experiments in which we further increased the goal distance from 25 to 100 steps, with a maximum planning budget of 125 steps (Table 3). CTWM also outperformed LeWM across environments in this harder setting, and the margin widened considerably. For example, LeWM collapses on Two-Room (17.3%), PushT (15.3%) and Pointmaze (28.2%), whereas CTWM retains near-ceiling performance on Two-Room (98.3%), Pointmaze (98.8%) and Reacher (98.2%), while success rate on PushT decreased for both algorithms. These results suggest that the benefits of a commute-time-preserving geometry become more pronounced as the goal moves further away. Thus, CTWM provides better task-agnostic rep-

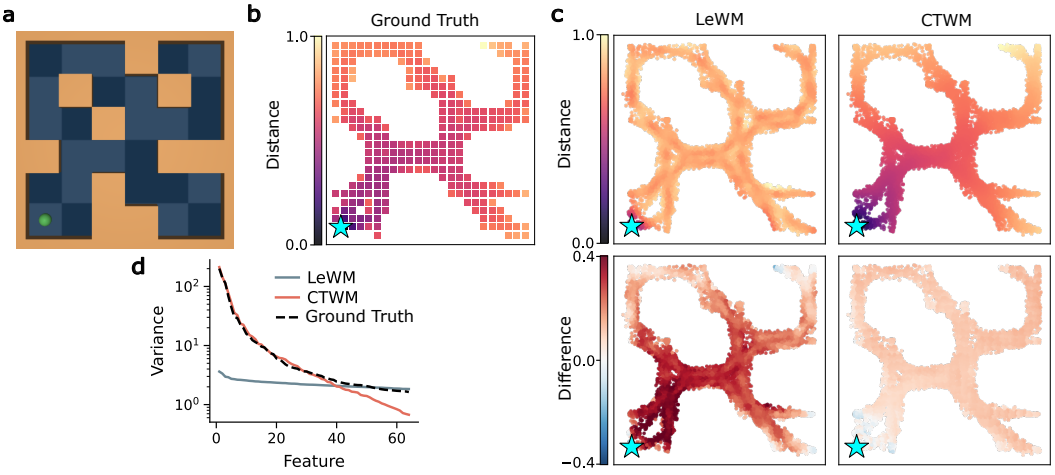


Figure 4: **Commute-time distance and variance scaling on Pointmaze.** **(a)** The view of the Pointmaze environment. **(b)** Ground truth commute-time distance from the blue star to any other point on the map computed from the discretized continuous state space. **(c)** Top row: Relative L_2 distance (normalised) in the embedding space of LeWM (left) and CTWM (right). Bottom row: difference to the ground-truth commute-time distance. CTWM reproduces the ground-truth distance more closely. **(d)** Variance of the first 64 latent features for both models. CTWM exhibits the $\lambda_k^{-1/2}$ decay predicted by Corollary 1.2 (dashed line), whereas LeWM produces a flatter spectrum.

resentations for planning in continuous environments, especially when faced with bottlenecks and higher-dimensional state spaces (see also Appendix E.1). Ultimately, we confirmed that these advantages also translate to gradient-based planning strategies (see Appendix E.2).

4.1.2 Commute-time geometry in continuous spaces

Finally, we tested whether CTWM's planning success correlates with its ability to faithfully represent commute-time distance in one of the above continuous environments. To that end, we discretized the Pointmaze state space into a graph, computed the ground-truth commute-time distances from a given point (Fig. 4b), and compared them to the L_2 embedding space distances by LeWM and CTWM (Fig. 4c). CTWM reproduced the ground-truth picture more faithfully. Then, we quantified the variance projected onto the first 64 feature dimensions (Fig. 4d). CTWM exhibited the characteristic $\lambda_k^{-1/2}$ decay as predicted (cf. Corollary 1.2), whereas LeWM produced a flatter decay consistent with a more isotropic representation. Together, these results suggest that CTWM faithfully represents commute-time geometry in continuous environments learned from pixels.

5 Conclusion

We introduced CTWM, a self-supervised world model whose latent geometry is shaped for planning rather than probing. CTWM predicts latent displacements, regularized by the log-det of the representation's covariance, provably recovers the scaled Laplacian representation under deterministic dynamics and at the predictor's fixed point, so that L_2 distance in latent space equals commute-time distance on the environment's transition graph. This makes the learned representation directly usable by any planner that scores candidate trajectories by goal distance. Across six goal-reaching tasks spanning navigation and manipulation, CTWM matches or outperforms LeWM while using half the parameters. Beyond planning performance, it is striking that a purely self-supervised, reward-free objective recovers the geometry that classical graph theory identifies as optimal for planning in reversible environments. This raises the broader question of whether this correspondence is a quirk of our architecture, or a general principle. We suspect the latter. Since commute time depends only on the environment, not on any policy or reward, it is a natural target for any agent that, like Tolman's rats mapping their maze before any reward is introduced (Tolman, 1948), must build a map before it knows what that model will be used for.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the writing of the proof in Appendix C.3. and to implement parts of our experimental code. All proofs were independently derived and verified by the authors, and all AI-assisted code was reviewed and tested by at least two authors. We did not use generative AI tools to generate synthetic datasets, propose or refine hypotheses, design our research methodology or experiments, clean or reformat datasets, or interpret results; the remaining tasks requiring disclosure are not applicable to this work. Additionally, we used generative AI tools to improve the readability of author-written text, to identify related literature, and to format references. All suggested references were manually checked against their original sources, and all AI-assisted text was reviewed and edited by the authors. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the code to reproduce the torus experiments, together with the code used to compare against LeWM, built based on the original LeWM code, as a supplementary ZIP archive. All hyperparameter values we used are reported in Appendix A. The assumptions and complete proofs of our theoretical results are given in Appendices C.2 and C.3. Figure 3 is included in the main text to show the evolution of performance over training; the corresponding final numerical results, with standard deviations across seeds, are reported in Appendix 3. Regarding the datasets: the archive includes the resized 14×14 MNIST images used to render the torus observations, and the continuous environments are those of the LeWM benchmark which can be downloaded here: https://huggingface.co/collections/quentinll/lewm. For the two additional benchmark environments, the supplementary code documents the data generation procedure, and we will release the generated datasets upon publication to remove this step.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. Td-jepa: Latent-predictive representations for zero-shot reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 36211–36244, 2026.

Randall Balestriero and Yann LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *Advances in Neural Information Processing Systems*, 35:26671–26685, 2022.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14, 2001.

Åke Björck and Gene H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973.

Maxime Burchi and Radu Timofte. Mudreamer: Learning predictive world models without reconstruction, 2024.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15745–15753. IEEE, 2021.

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

Marc Peter Deisenroth and Carl Edward Rasmussen. Pilco: a model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 465–472. Omnipress, 2011.

Fei Deng, Ingook Jang, and Sungjin Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations, 2021.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

Jesse Farebrother, Matteo Pirotta, Andrea Tirinzoni, Rémi Munos, Alessandro Lazaric, and Ahmed Touati. Temporal difference flows. *arXiv preprint arXiv:2503.09817*, 2025.

Jesse Farebrother, Matteo Pirotta, Andrea Tirinzoni, Marc G Bellemare, Alessandro Lazaric, and Ahmed Touati. Compositional planning with jumpy world models. *arXiv preprint arXiv:2602.19634*, 2026.

Diego Gomez, Michael Bowling, and Marlos C Machado. Proper laplacian representation learning. In *International Conference on Learning Representations*, volume 2024, pp. 28911–28934, 2024.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

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

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models, 2022.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Manu Srinath Halvagal, Axel Laborieux, and Friedemann Zenke. Implicit variance regularization in non-contrastive SSL. *NeurIPS*, 2023.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Michael Hauri and Friedemann Zenke. Dreamer-cdp: Improving reconstruction-free world models via continuous deterministic representation prediction. *arXiv preprint arXiv:2603.07083*, 2026.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Yehuda Koren. On spectral graph drawing. In *International Computing and Combinatorics Conference*, pp. 496–508. Springer, 2003.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Marlos C Machado, Marc G Bellemare, and Michael Bowling. A laplacian framework for option discovery in reinforcement learning. In *International conference on machine learning*, pp. 2295–2304. PMLR, 2017.

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

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

Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8(10), 2007.

Fabian A Mikulasch and Friedemann Zenke. Understanding self-supervised learning via latent distribution matching. *arXiv preprint arXiv:2605.03517*, 2026.

Vivek Myers, Bill Zheng, Benjamin Eysenbach, and Sergey Levine. Offline goal-conditioned reinforcement learning with quasimetric representations. *Advances in Neural Information Processing Systems*, 38:19654–19679, 2026.

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

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

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, volume 2025, pp. 94937–94982, 2025.

Seohong Park, Deepinder Mann, and Sergey Levine. Dual goal representations. In *International Conference on Learning Representations*, volume 2026, pp. 153307–153324, 2026.

Huaijun Qiu and Edwin R Hancock. Clustering and embedding using commute times. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1873–1890, 2007.

Stefan Richthofer and Laurenz Wiskott. Predictable feature analysis. In *2015 IEEE 14th international Conference on machine learning and applications (ICMLA)*, pp. 190–196. IEEE, 2015.

Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*, volume 133. Springer, 2004.

Tankred Saanum, Peter Dayan, and Eric Schulz. Simplifying latent dynamics with softly state-invariant world models. *Advances in Neural Information Processing Systems*, 37:38355–38382, 2024.

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*, 588 (7839):604–609, December 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-03051-4.

Dikshant Shehmar, Matthew Schlegel, Matthew E Taylor, and Marlos C Machado. Laplacian representations for decision-time planning. *arXiv preprint arXiv:2602.05031*, 2026.

Ravid Shwartz-Ziv, Randall Balestriero, Kenji Kawaguchi, Tim GJ Rudner, and Yann LeCun. An information theory perspective on variance-invariance-covariance regularization. *Advances in neural information processing systems*, 36:33965–33998, 2023.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. doi: 10.1038/nature16961.

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

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

Henning Sprekeler. On the relation of slow feature analysis and laplacian eigenmaps. *Neural computation*, 23(12):3287–3302, 2011.

Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine Learning Proceedings 1990*, pp. 216–224. 1990. ISBN 978-1-55860-141-3. doi: https://doi.org/10.1016/B978-1-55860-141-3.50030-4. URL https://www.sciencedirect.com/science/article/pii/B9781558601413500304.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pp. 10268–10278. PMLR, 2021.

Edward C Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4):189–208, 1948. doi: 10.1037/h0061626.

Edward C Tolman and Charles H Honzik. Introduction and removal of reward, and maze performance in rats. *University of California Publications in Psychology*, 4:257–275, 1930.

Jyothir S V, Siddhartha Jalagam, Yann LeCun, and Vlad Sobal. Gradient-based planning with world models, 2023.

Harm van Seijen, Hadi Nekoei, Evan Racah, and Sarath Chandar. The loca regret: A consistent metric to evaluate model-based behavior in reinforcement learning, 2020.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Kaixin Wang, Kuangqi Zhou, Qixin Zhang, Jie Shao, Bryan Hooi, and Jiashi Feng. Towards better laplacian representation in reinforcement learning with generalized graph drawing. In *International Conference on Machine Learning*, pp. 11003–11012. PMLR, 2021.

Kaixin Wang, Kuangqi Zhou, Jiashi Feng, Bryan Hooi, and Xinchao Wang. Reachability-aware laplacian representation in reinforcement learning. *arXiv preprint arXiv:2210.13153*, 2022.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning*, pp. 36411–36430. PMLR, 2023.

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim GJ Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning. *arXiv preprint arXiv:2603.12231*, 2026.

Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in rl: Learning representations with efficient approximations. *arXiv preprint arXiv:1810.04586*, 2018.

Wenjun Xiao and Ivan Gutman. Resistance distance and laplacian spectrum. *Theoretical chemistry accounts*, 110(4):284–289, 2003.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pp. 12310–12320. PMLR, 2021.

Kai Zhao, Dongliang Nie, Yuchen Lin, Zhehan Luo, Yixiao Gu, Deng-Ping Fan, and Dan Zeng. Sub-jepa: Subspace gaussian regularization for stable end-to-end world models. *arXiv preprint arXiv:2605.09241*, 2026.

Bill Chunyuan Zheng, Vivek Myers, Benjamin Eysenbach, and Sergey Levine. Multistep quasimetric learning for scalable goal-conditioned reinforcement learning. *arXiv preprint arXiv:2511.07730*, 2025.

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

Jiachen Zhu, Katrina Evtimova, Yubei Chen, Ravid Shwartz-Ziv, and Yann LeCun. Variance-covariance regularization improves representation learning. *arXiv preprint arXiv:2306.13292*, 2023.

A HYPERPARAMETERS

	Hyperparameter	Value
Architecture	encoder ϕ	ViT-Tiny (5.5 M)
	projector	MLP with LayerNorm
	predictor T	Transformer architecture of (Maes et al., 2026b) of width D (3.3 M)
Data	image size	224×224
	context length H	3
	frame skip k	5
	prediction steps	1
	embedding dimension D	64
Optimisation	optimiser	AdamW
	weight decay	0
	learning rate	5×10^{-5}
	batch size	32
	epochs	10
Loss	β	0.1
	γ	1.0
	ϵ	10^{-6}
Planning	solver	CEM: 300 samples, 30 iterations, 30 elites, σ scale 1.0
	horizon	5 blocks of $k = 5$ actions
	replanning interval	5 blocks
	goal offset / step budget	25 / 50 environment steps
	evaluation episodes	200

Table 1: Main hyperparameters. (Green) Changes from LeWM.

B REGULARIZERS

B.1 ORTHONORMALITY

Wu et al. (2018) constrain the coordinates of ϕ to be orthonormal under the data distribution, enforced by the soft penalty

$$Reg(\phi) = \frac{\delta}{2}\Big(\Big\|\tfrac{1}{N} Z^\top Z - I_d\Big\|_F^2 + \Big\|\tfrac{1}{N} Z'^\top Z' - I_d\Big\|_F^2\Big), \tag{14}$$

which drives the second moments of the representations toward the identity.

B.2 VCREG

VICReg (Bardes et al., 2021) is a common SSL regularization methods and also motivates the loss of Sobal et al. (2026). The variance term $\mathcal{L}_v(Z, Z') = v(Z) + v(Z')$ with

$$v(Z) = \frac{1}{d}\sum_{j=1}^{d} \max\Big(0,\ \tau - \sqrt{\mathrm{Var}(z^j) + \epsilon}\Big) \tag{15}$$

pushes the standard deviation of every coordinate above the threshold τ. The covariance term $\mathcal{L}_c(Z, Z') = c(Z) + c(Z')$ with $c(Z) = \frac{1}{d}\sum_{i \neq j} \mathrm{Cov}(Z)_{ij}^2$ decorrelates the coordinates. We use only these two terms, i.e. VCReg (Zhu et al., 2023) and do not consider the invariance term as this could be seen to be implemented through the main loss:

$$Reg(\phi) = \alpha_1 \mathcal{L}_c(Z, Z') + \alpha_2 \mathcal{L}_v(Z, Z'), \tag{16}$$

which can be seen as a relaxation of the orthonormality constraint: coordinates are decorrelated and each standard deviation must reach τ, but is otherwise free.

C PROOFS

C.1 CONVERGENCE OF EQUATION 5

Lemma 1. *If learning achieves the optimum of the CTWM loss function (equation 5)*

$$\mathcal{L}(\phi, T) = \mathbb{E}_{(s,a,s')}[\|\Delta(s,s') - T(\phi(s),a)\|^2 + \beta\|T(\phi(s),a)\|^2] + Reg(\phi), \tag{17}$$

and if the dynamics are deterministic and regularization $Reg(\phi)$ is an orthonormality constraint, then the latent states z_t converge to the Laplacian representation.

Proof. For simplicity, let $\Delta := \phi(s') - \phi(s)$, and define the conditional mean displacement as $\mu := \mathbb{E}[\Delta|s,a] = \mathbb{E}_{s'\sim P(\cdot|s,a)}[\phi(s')] - \phi(s)$. In the following we also drop the arguments of the predictor $T(\phi(s),a)$. Using the multivariate bias-variance decomposition

$$\mathbb{E}\left[\|\Delta - T\|^2|s,a\right] = \mathrm{tr}\big(\mathrm{Cov}(\Delta|s,a)\big) + \|\mu - T\|^2$$

the conditional objective becomes

$$\mathcal{L}(\phi, T) = \mathrm{tr}\big(\mathrm{Cov}(\Delta|s,a)\big) + \|\mu - T\|^2 + \beta\|T\|^2 + Reg(\phi). \tag{18}$$

For $\beta \geq 0$, this objective is strictly convex in T. Its gradient is $\nabla_T\mathcal{L}(\phi,T) = 2(T-\mu) + 2\beta T$, so the unique minimizer satisfies $\nabla_T\mathcal{L}(\phi,T) = 0$. Therefore,

$$T^\star(s,a) = \frac{\mu(s,a)}{1+\beta}. \tag{19}$$

Substituting $T^\star$ into the objective gives

$$\begin{aligned}
\mathcal{L}(\phi, T^\star) &= \mathbb{E}_{(s,a)}\left[\mathrm{tr}\big(\mathrm{Cov}(\Delta|s,a)\big) + \left\|\mu - \frac{\mu}{1+\beta}\right\|^2 + \beta\left\|\frac{\mu}{1+\beta}\right\|^2\right] + Reg \\
&= \mathbb{E}_{(s,a)}\left[\mathrm{tr}\big(\mathrm{Cov}(\Delta|s,a)\big)\right] + \frac{\beta^2}{(1+\beta)^2}\mathbb{E}_{(s,a)}[\|\mu\|^2] + \frac{\beta}{(1+\beta)^2}\mathbb{E}_{(s,a)}[\|\mu\|^2] + Reg \\
&= \mathbb{E}_{(s,a)}\left[\mathrm{tr}\big(\mathrm{Cov}(\Delta|s,a)\big)\right] + \frac{\beta(1+\beta)}{(1+\beta)^2}\mathbb{E}_{(s,a)}[\|\mu\|^2] + Reg \\
&= \mathbb{E}_{(s,a)}\left[\mathrm{tr}\big(\mathrm{Cov}(\Delta|s,a)\big)\right] + \frac{\beta}{1+\beta}\mathbb{E}_{(s,a)}[\|\mu(s,a)\|^2] + Reg.
\end{aligned}$$

Thus, the final objective can be written as

$$\boxed{\mathcal{L}(\phi, T^\star) = \mathbb{E}_{(s,a)}\left[\mathrm{tr}\big(\mathrm{Cov}(\Delta|s,a)\big)\right] + \frac{\beta}{1+\beta}\mathbb{E}_{(s,a)}[\|\mu(s,a)\|^2] + Reg.} \tag{20}$$

If the dynamics are deterministic, i.e., $s' = F(s,a)$, the covariance is 0 and the conditional mean displacement $\mu(s,a) = \Delta$, the objective simplifies to:

$$\mathcal{L}(\phi, T^\star) = \frac{\beta}{1+\beta}\mathbb{E}_{(s,s')}[\|\phi(s') - \phi(s)\|^2] + Reg. \tag{21}$$

If the regularization implements an orthonormality constraint, this is the Graph-drawing objective of Koren (2003); Wu et al. (2018). As proven previously, learned representations under this loss converge to the Laplacian representation up to a rotation (Wang et al., 2021; Gomez et al., 2024). □

C.2 SCALING OF THE REPRESENTATION UNDER DIFFERENT REGULARIZERS (HEURISTICS)

In the following we derive heuristics for the scaling of representations under different regularizers, by assuming that representations align with the eigenvectors of the Laplacian—even though not all regularizers guarantee this. We first consider the objective with an arbitrary regularizer, and later investigate orthonormality, VCReg and log-det as three particular instances.

Write $c := \beta/(1+\beta)$ and assume the representation is aligned with the eigenbasis, i.e., each coordinate is a scaled eigenvector, $\phi_k = s_k u_{\pi(k)}$ for distinct indices $\pi(1), \dots, \pi(d)$, where $d = \dim \mathcal{Z}$ and π a permutation. Since the eigenvectors satisfy $\mathbb{E}[u_k u_\ell] = \delta_{k\ell}$ and $\mathbb{E}[u_k] = 0$, we have $\mathrm{var}(\phi_k) = s_k^2$ and $\Phi = \mathrm{diag}(s_1^2, \dots, s_d^2)$. Applying Von Luxburg (2007, Prop. 1) to each coordinate and by the eigenvalue problem $Lu_\ell = \lambda_\ell D u_\ell$, the predictive term can be rewritten as $s_k s_l u_k^\top L u_\ell = s_k s_l \lambda_\ell \delta_{k\ell}$ which implies:

$$\frac{\beta}{1+\beta}\,\mathbb{E}_{(s,s')}\big[\|\phi(s') - \phi(s)\|^2\big] = 2c \sum_{k=1}^{d} \lambda_{\pi(k)}\, s_k^2, \tag{22}$$

so the objective separates across directions: each contributes a cost $2c\lambda_{\pi(k)} s_k^2$, increasing in both the eigenvalue and the variance, against whatever the regularizer contributes. The regularizer alone therefore determines the scaling s_k. In each case, the selected eigendirections are the d slowest, $\{\pi(1), \dots, \pi(d)\} = \{1, \dots, d\}$, since equation 22 is increasing in λ.

Orthonormality. The constraint $\Phi = I_d$ fixes $s_k = 1$ directly:

$$\mathrm{var}(\phi_k^\star) = 1. \tag{23}$$

The variance is the same in every direction, and the representation is the unscaled Laplacian representation $(u_k)_{k=1}^d$. The general setup is studied in Koren (2003).

VCReg. In this case, the objective is $\sum_{k=1}^d \big[2c\lambda_k s_k^2 + \alpha \max(0, \tau - s_k)\big]$. For $s_k > \tau$ the solution is τ. For $s_k < \tau$ the derivative is $4c\lambda_k s_k - \alpha$, which gives a minimum at $s_k = \alpha/(4c\lambda_k)$. In other words,

$$\mathrm{var}(\phi_k^\star) = \begin{cases} \tau^2, & \lambda_k \le \dfrac{\alpha}{4c\tau}, \\ \dfrac{\alpha^2}{16c^2\lambda_k^2}, & \lambda_k > \dfrac{\alpha}{4c\tau}, \end{cases} \tag{24}$$

Fast directions decays as λ_k^{-2} whereas the slow ones are constant. Note that Balestriero & LeCun (2022) introduce a surrogate coinciding with the soft penalty of Appendix B.1, which does not recover the commute-time scaling either.

Log-det. For the log-det variant, the objective is $\sum_{k=1}^d \big[2c\lambda_k s_k^2 - \gamma \log s_k\big]$. The derivative for each term is $4c\lambda_k s_k - \gamma/s_k$, so the unique minimum is at

$$\mathrm{var}(\phi_k^\star) = \frac{\gamma}{4c\lambda_k} = \frac{\gamma(1+\beta)}{4\beta\,\lambda_k}. \tag{25}$$

which is the scaling ($\propto \lambda_k^{-1}$ equation 8) we are interested in for the scaled Laplacian representation. A stronger result is provided in the next section for the log-det where the assumption of alignement with the eigenbasis is removed.

C.3 Proof of theorem 1

Theorem 1 (Restated). *Let $(\lambda_k, u_k)_{k\ge 0}$ denote the eigenpairs of the Laplacian, ordered as*

$$0 = \lambda_0 < \lambda_1 \le \cdots \le \lambda_{n-1},$$

and assume an eigengap $\lambda_d < \lambda_{d+1}$, where $d = \dim \mathcal{Z}$. Let

$$\psi(s) = \begin{pmatrix} \lambda_1^{-1/2} u_1(s) \\ \vdots \\ \lambda_d^{-1/2} u_d(s) \end{pmatrix}$$

be the rank-d scaled Laplacian representation.

Under deterministic dynamics, consider the reduced CTWM objective following Lemma 1

$$\mathcal{L}(\phi) = \frac{\beta}{1+\beta}\mathbb{E}_{(s,s')}\left[\|\phi(s') - \phi(s)\|_2^2\right] - \frac{\gamma}{2}\log\det\Sigma_\phi, \qquad \Sigma_\phi := \mathrm{Cov}_\rho[\phi(s)], \tag{26}$$

where ρ is the stationary state distribution. Then every minimizer is, up to an orthogonal transformation and a translation,

$$\phi^\star(s) = \sqrt{\frac{\gamma(1+\beta)}{4\beta}}\, O\psi(s) + b, \qquad O \in \mathsf{O}(d).$$

Proof. The proof proceeds in four steps. We first reduce the problem to a matrix optimization (Step 1), then factor the coefficient matrix into the subspace R spanned by the features and the covariance Σ that scales it, rewriting the loss in terms of both (Step 2). We then minimize over Σ for a fixed subspace (Step 3), and substituting the result back leaves a problem in R alone, which we solve with the Poincaré separation theorem (Step 4).

Step 0: Notation. Define $c := \beta/(1+\beta)$. Since both terms in the objective are invariant under translation of the representation, we may assume without loss of generality that $\mathbb{E}_\rho[\phi(s)] = 0$.

Let $u_1, \dots, u_{n-1}$ denote the nonconstant Laplacian eigenfunctions, normalized with respect to the stationary state distribution ρ such that $\mathbb{E}_\rho[u_k(s)u_\ell(s)] = \delta_{k\ell}$. Collect them in

$$u(s) = \begin{pmatrix} u_1(s) \\ \vdots \\ u_{n-1}(s) \end{pmatrix}, \qquad \Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_{n-1}).$$

Because the u_k form an orthonormal basis of the centered functions on the finite state space, any centered d-dimensional representation can be expanded as

$$\phi(s) = Au(s), \qquad A \in \mathbb{R}^{d\times(n-1)}.$$

Step 1: Reduction to a matrix optimization. By orthonormality of the Laplacian eigenfunctions,

$$\Sigma_\phi = \operatorname{Cov}_\rho[\phi(s)] = AA^\top.$$

We will now rewrite equation 26 in terms of A. We first apply to its first term Von Luxburg (2007, Prop. 1) to each coordinate:

$$\mathbb{E}\|\phi(s') - \phi(s)\|^2 = \sum_j \mathbb{E}(\phi_j(s') - \phi_j(s))^2 = 2\sum_j \phi_j^\top L \phi_j.$$

In addition, if we use the definition of the generalized eigenproblem $Lu_\ell = \lambda_\ell D u_\ell$, we have:

$$u_k^\top L u_\ell = \lambda_\ell u_k^\top D u_\ell = \lambda_\ell \mathbb{E}[u_k(s)u_\ell(s)] = \lambda_\ell \delta_{k\ell}.$$

Combining the two last equations leads to:

$$\mathbb{E}\|\phi(s') - \phi(s)\|^2 = 2\sum_j \sum_{k,\ell} A_{jk}A_{j\ell}\, u_k^\top L\, u_\ell = 2\sum_j \sum_k \lambda_k A_{jk}^2 = 2\operatorname{tr}(A\Lambda A^\top).$$

with $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_{n-1})$. The objective therefore becomes

$$\mathcal{L} = 2c\operatorname{tr}(A\Lambda A^\top) - \tfrac{\gamma}{2}\log\det(AA^\top). \tag{27}$$

Step 2: Separate the represented subspace from its scaling. Let

$$\Sigma := AA^\top \succ 0, \qquad R := \Sigma^{-1/2}A.$$

Then

$$RR^\top = \Sigma^{-1/2}AA^\top\Sigma^{-1/2} = I_d,$$

and

$$A = \Sigma^{1/2}R.$$

This decomposition separates two properties of the representation: R determines the d-dimensional subspace represented by ϕ, whereas Σ determines the covariance, and hence the scaling, within that subspace.

Substituting $A = \Sigma^{1/2} R$ into equation 27 gives

$$\mathcal{L}(\Sigma, R) = 2c \operatorname{tr}\left(\Sigma^{1/2} R \Lambda R^\top \Sigma^{1/2}\right) - \frac{\gamma}{2} \log\det \Sigma \tag{28}$$

$$= 2c \operatorname{tr}\left(\Sigma R \Lambda R^\top\right) - \frac{\gamma}{2} \log\det \Sigma. \tag{29}$$

Define the Laplacian restricted to the selected subspace as

$$M_R := R \Lambda R^\top.$$

Since $\lambda_k > 0$ for all $k \geq 1$ and R has full row rank, $M_R \succ 0$.

Step 3: Optimal covariance for a fixed subspace. For fixed R, equation 29 is minimized over $\Sigma \succ 0$. Using

$$\nabla_\Sigma \operatorname{tr}(\Sigma M_R) = M_R, \qquad \nabla_\Sigma \log\det \Sigma = \Sigma^{-1},$$

the stationary condition is

$$2cM_R - \frac{\gamma}{2}\Sigma^{-1} = 0.$$

Therefore,

$$\Sigma_R^\star = \frac{\gamma}{4c} M_R^{-1} = \frac{\gamma}{4c}\left(R \Lambda R^\top\right)^{-1}. \tag{30}$$

Thus, for any fixed spectral subspace, the optimal representation covariance is proportional to the inverse Laplacian restricted to that subspace.

Substituting equation 30 back into the objective gives

$$\mathcal{L}(\Sigma_R^\star, R) = 2c \operatorname{tr}\left(\frac{\gamma}{4c} M_R^{-1} M_R\right) - \frac{\gamma}{2} \log\det\left(\frac{\gamma}{4c} M_R^{-1}\right)$$

$$= \frac{\gamma d}{2} - \frac{\gamma d}{2} \log \frac{\gamma}{4c} + \frac{\gamma}{2} \log\det M_R.$$

The first two terms do not depend on R. Hence optimizing the subspace is equivalent to

$$\min_{R R^\top = I_d} \log\det(R \Lambda R^\top).$$

Step 4: The optimal subspace contains the slowest modes. Let

$$\mu_1 \leq \cdots \leq \mu_d$$

denote the eigenvalues of $R \Lambda R^\top$. By the Poincaré separation theorem,

$$\lambda_k \leq \mu_k, \qquad k = 1, \ldots, d.$$

Therefore,

$$\det(R \Lambda R^\top) = \prod_{k=1}^{d} \mu_k \geq \prod_{k=1}^{d} \lambda_k.$$

Equality is attained when the row space of R is the span of the first d Laplacian eigendirections. Because $\lambda_d < \lambda_{d+1}$, this subspace is unique. Hence, up to an orthogonal transformation $O \in \mathsf{O}(d)$,

$$R^\star = O \begin{bmatrix} I_d & 0 \end{bmatrix}.$$

Writing

$$\Lambda_d = \operatorname{diag}(\lambda_1, \ldots, \lambda_d),$$

we consequently have

$$M_{R^\star} = R^\star \Lambda R^{\star\top} = O \Lambda_d O^\top.$$

Equation 30 then gives

$$\Sigma^\star = \frac{\gamma}{4c} O \Lambda_d^{-1} O^\top.$$

One corresponding coefficient matrix is

$$A^\star = \sqrt{\frac{\gamma}{4c}}\, O \begin{bmatrix} \Lambda_d^{-1/2} & 0 \end{bmatrix}.$$

Thus

$$\phi^\star(s) = A^\star u(s) = \sqrt{\frac{\gamma}{4c}}\, O\psi(s).$$

Finally, substituting $c = \beta/(1+\beta)$ yields

$$\phi^\star(s) = \sqrt{\frac{\gamma(1+\beta)}{4\beta}}\, O\psi(s).$$

Restoring the arbitrary translation b gives

$$\phi^\star(s) = b + \sqrt{\frac{\gamma(1+\beta)}{4\beta}}\, O\psi(s)$$

which corresponds to equation 10. □

C.4 Proof of Corollary 1.2

Proof. By definition of the eigenvectors $(u_k)_{k>0}$ of the Laplacian $\mathbb{E}[u_k(s)u_\ell(s)] = \delta_{k\ell}$ (Belkin & Niyogi, 2001). Moreover $u_0 \equiv 1$ is itself an eigenvector, so orthogonality to it gives $\mathbb{E}[u_k] = \mathbb{E}[u_k \cdot 1] = 0$ for $k \geq 1$. Hence $\mathrm{Cov}(u_k, u_\ell) = \mathbb{E}[u_k(s)u_\ell(s)] - \mathbb{E}[u_k(s)]\mathbb{E}[u_\ell(s)] = \delta_{k\ell}$, and since $\psi_k = \lambda_k^{-1/2} u_k$,

$$\mathrm{Cov}(\phi^\star) = \frac{\gamma(1+\beta)}{4\beta}\,\mathrm{Cov}(\psi) = \frac{\gamma(1+\beta)}{4\beta}\Lambda_d^{-1}. \tag{31}$$

□

D Smoothness constraint

D.1 Smoothness with next-state prediction

Instead of using residual prediction as in 5, we could reparametrize it into a mathematically equivalent next-state prediction loss. Let's denote with $N(\phi(s), a)$ the network to predict the next state embedding starting form the current state embedding and the action. We can write the following loss:

$$\mathcal{L}(\phi, T) = \mathbb{E}_{(s,a,s')}[\|\phi(s') - N(\phi(s), a)\|^2 + \beta\|N(\phi(s), a) - \phi(s)\|^2] + Reg(\phi). \tag{32}$$

Now, if we substitute $N(\phi(s), a) = \phi(s) + T(\phi(s), a)$ we get:

$$\mathcal{L}(\phi, T) = \mathbb{E}_{(s,a,s')}[\|\phi(s') - \phi(s) - T(\phi(s), a)\|^2 + \beta\|\phi(s) + T(\phi(s), a) - \phi(s)\|^2] + Reg(\phi).$$

By simplifying we obtain:

$$\mathcal{L}(\phi, T) = \mathbb{E}_{(s,a,s')}[\|\phi(s') - \phi(s) - T(\phi(s), a)\|^2 + \beta\|T(\phi(s), a)\|^2] + Reg(\phi). \tag{33}$$

Eventually, denoting with $\Delta(s, s') = \phi(s') - \phi(s)$, we have:

$$\mathcal{L}(\phi, T) = \mathbb{E}_{(s,a,s')}[\|\Delta(s, s') - T(\phi(s), a)\|^2 + \beta\|T(\phi(s), a)\|^2] + Reg(\phi). \tag{34}$$

This is the equation 5.

β	β_{est}
0.005	0.0074
0.1	0.108
0.25	0.26
0.5	0.52
1.0	1.04

Table 2: Comparison of β and β_{est} on PushT.

D.2 Planning with beta

The fixed point condition (equation 19) implies that our planning method should rely on:

$$\phi(s_{t+1}) = \phi(s_t) + (1+\beta)T(\phi(s_t), a_t). \tag{35}$$

We trained the model with different beta values and estimated the β_{est} that would minimize the scaling error from the predictor for each of them using Ordinary Least-Squares (OLS):

$$\phi(s_{t+1}) - \phi(s_t) = (1+\beta_{\text{est}})T(\phi(s_t), a_t) \quad . \tag{36}$$

The results (Table 2) show that the estimated scaling for the predictor is close to the one computed in equation 19 on the PushT environment.

E Planning

E.1 Planning numerical results

Success rate	Reacher	Cube	Two-Room	PushT	Scene	Pointmaze
Random	$13.0 \pm 2.3\%$	$45.2 \pm 0.9\%$	$24.3 \pm 2.7\%$	$4.3 \pm 2.1\%$	$49.8 \pm 3.1\%$	$14.3 \pm 1.7\%$
LeWM	$83.3 \pm 2.1\%$	$64.8 \pm 1.3\%$	$88.8 \pm 1.6\%$	$91.5 \pm 1.1\%$	$69.3 \pm 2.6\%$	$68.5 \pm 1.8\%$
CTWM	$82.3 \pm 2.6\%$	$89.8 \pm 1.0\%$	$100.0 \pm 0.0\%$	$92.2 \pm 0.6\%$	$82.3 \pm 1.4\%$	$100.0 \pm 0.0\%$
Random-100	$7.3 \pm 1.9\%$	$45.5 \pm 2.5\%$	$1 \pm 0.4\%$	$0.3 \pm 0.2\%$	$24.2 \pm 1.7\%$	$11.3 \pm 1.0\%$
LeWM-100	$81 \pm 0.8\%$	$61.7 \pm 1.4\%$	$17.3 \pm 0.8\%$	$15.3 \pm 1.4\%$	$40.3 \pm 2.5\%$	$28.2 \pm 1.2\%$
CTWM-100	$98.2 \pm 0.8\%$	$73.0 \pm 1.2\%$	$98.3 \pm 0.9\%$	$26.5 \pm 2.6\%$	$57.5 \pm 1.6\%$	$98.8 \pm 0.2\%$

Table 3: Planning success rate for each model trained on three seeds after ten epochs using CEM for the standard setting by Maes et al. (2026a) with a goal distance of 25 steps and planning budget of 50 steps (top half) and for an extended goal distance of 100 steps with planning budget of 125 steps (bottom half). A baseline floor is given by the random policy for each setting and environment.

E.2 Planning with first-order methods

Success rate	Reacher	Cube	Two-Room	PushT	Scene	Pointmaze
Random	$13.0 \pm 2.3\%$	$45.2 \pm 0.9\%$	$24.3 \pm 2.7\%$	$4.3 \pm 2.1\%$	$49.8 \pm 3.1\%$	$14.3 \pm 1.7\%$
LeWM	$73.7 \pm 2.7\%$	$60.2 \pm 0.5\%$	$93.7 \pm 2.5\%$	$86.5 \pm 1.2\%$	$66.7 \pm 2.4\%$	$88.3 \pm 1.2\%$
CTWM	$77.0 \pm 3.3\%$	$78.2 \pm 2.1\%$	$98.7 \pm 0.8\%$	$87.5 \pm 1.1\%$	$72.8 \pm 2.5\%$	$100.0 \pm 0.0\%$

Table 4: Planning success rate for each model trained on three seeds after ten epochs using Adam. A floor is given by the random policy for each environment.

F Subspace similarity

The subspace similarity is measured using the canonical angles (principal angles) method, a mathematical technique used to measure the alignment and distance between two linear subspaces. Let $\Phi \in \mathbb{R}^{n \times d}$ collect the learned embeddings of the n graph nodes, and let $U \in \mathbb{R}^{n \times d}$ collect the

Laplacian eigenvectors $u_1, \ldots, u_d$, excluding the constant eigenvector u_0. We first subtract the mean embedding from Φ, since the Laplacian eigenvectors are orthogonal to the constant vector and the learned embeddings are only defined up to a translation. We then take the left singular vectors of the centered Φ as an orthonormal basis A of its column space, and similarly obtain B from U. The singular values $\sigma_1 \geq \cdots \geq \sigma_d$ of $A^\top B$ are the cosines of the principal angles between the two subspaces (Björck & Golub, 1973), and we report their mean, $\frac{1}{d}\sum_i \sigma_i \in [0, 1]$, which equals 1 for identical and 0 for orthogonal subspaces. The same procedure is applied to all methods. This measure is invariant to rotations within degenerate eigenspaces, which is essential on the torus, and to any rescaling of the embedding; eigenvalue-dependent scaling is therefore assessed separately (Fig. 2c).