MODULAR RECURRENT SCAFFOLDS FOR COMPOSITIONAL WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models often learn representations and transition dynamics jointly, entangling perception with task structure and limiting compositional generalization. We introduce a hippocampal-inspired world model built on pretrained recurrent modules whose dynamics are frozen during downstream learning. Their multi-scale periodic codes jointly disambiguate states that would be aliased within any individual module, providing a stable and structured latent coordinate system. A lightweight learned interface binds observations and actions to this scaffold, allowing the model to recover the topology of new task spaces with substantially fewer trainable parameters than end-to-end alternatives. The resulting representation decomposes task-relevant factors across modular dynamical subspaces and supports their recombination in previously unseen configurations. By exploiting this learned task-space structure, the model improves out-of-distribution compositional generalization and maintains more accurate long-horizon rollouts. The same structured latent geometry also enables goal-conditioned navigation and planning from state differences. These results suggest that pretrained multiscale grid-like dynamics can serve as a reusable scaffold for learning structured, compositional world models efficiently.

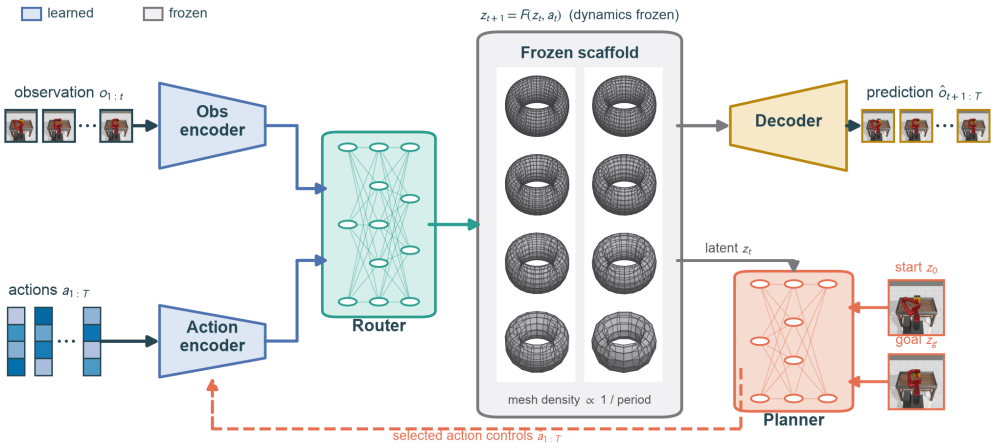


Figure 1: Overview of modular recurrent scaffolds. A bank of recurrent modules is pretrained on two-dimensional path integration and then frozen. For each new domain, a learned router binds encoded observations and embedded action identifiers to the frozen scaffold, the scaffold is rolled forward under action controls, and a goal-conditioned planner operates on latent state differences. Only the sensory encoder/decoder, router, action interface, and planner are learned downstream; the recurrent dynamics are never updated.

1 INTRODUCTION

World models compress high-dimensional observations into latent states and recurrently evolve those states to support prediction and control. Modern latent world models learn the encoder and the transition function jointly (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2025), so that perception and dynamics are entangled and the geometry of the latent space is determined implicitly by the

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.

training task. While large language models let useful dynamics "emerge" through next-token prediction and later shaping by supervised and reinforcement fine-tuning (Brown et al., 2020; Ouyang et al., 2022), world models typically require substantial task-specific training and still struggle to capture the hidden or intrinsic structure of their data.

This coupling has concrete consequences. Because natural observations are approximately low-rank, the embedded manifold is thin (Fefferman et al., 2016), and the states produced by an encoder occupy a restricted region of the ambient latent space. Small one-step errors then push an autoregressive rollout off this manifold, where neither the transition model nor the decoder is well constrained, and errors compound over long horizons (Talvitie, 2017; Janner et al., 2019). Related work mitigates this by regularizing the latent space—for example with Lipschitz penalties, contrastive objectives, or architectural factorization (Miyato et al., 2018; van den Oord et al., 2018; Kipf et al., 2020)—but such fixes are often costly in performance and require an appropriate metric or distance that is itself difficult to define.

The natural world, however, is not unstructured: tasks and knowledge are usually organized by underlying regularities, and representation learning has succeeded in vision, audio, and language precisely by extracting reusable feature hierarchies (Bengio et al., 2013; He et al., 2016). This suggests that concise, structured dynamics could be learned once and then reused, rather than relearned for every new observation domain.

Neuroscience offers a concrete template. Hippocampal and entorhinal grid- and place-cell systems represent not only physical space but also abstract task, knowledge, and relational structure (Behrens et al., 2018; Whittington et al., 2022). The attractor dynamics of grid codes are low-dimensional and therefore tractable (Burak & Fiete, 2009), yet remain faithful over long horizons with limited drift. Such a representation is well suited to fast, structured *binding* between a messy observational space and a clean, rigorous representational space—a "hippo-space" that can serve as a sketchboard for storing, manipulating, and sampling structured states. Crucially, re-structuring information onto a clean scaffold may enable few-shot learning and compositional generalization by reusing a well-organized relational map, much as humans and other mammals learn from sparse and noisy observations with remarkable speed (Lake et al., 2017; Lake & Baroni, 2018).

We turn this idea into a concrete design principle: *freeze the dynamics and learn the interface*. First, we pretrain a bank of recurrent modules on two-dimensional path integration and permanently freeze their dynamics, obtaining a structured latent coordinate system with known geometry. Second, a modality-specific encoder produces a distributed representation of an observation, and a shared component router binds it to the phase ports of the frozen modules, yielding a valid scaffold state. Third, discrete action identifiers are embedded and passed through the same router to produce module-specific controls, so their meaning is inferred purely from predictive supervision. Finally, a goal-conditioned planner compares imagined and goal states in the frozen latent space and selects actions from latent-state differences. The downstream model never receives coordinates, factor labels, absolute state identifiers, or hand-crafted action vectors.

Contributions.

- We introduce a pretrained hippocampal-inspired scaffold whose multiscale recurrent modules disambiguate aliased states and provide stable, frozen latent dynamics (Section 3 and Section 4).
- We show that a lightweight interface binds new observations and actions to this fixed z-space, recovering structured task geometry with fewer trainable parameters and improving OOD compositional generalization (Section 5).
- We demonstrate that the recovered geometry supports accurate long-horizon rollouts and goal-conditioned planning across language, vision, and control, suggesting a transferable scaffold for world-model learning (Section 6).

2 Background and Related Work

Predictive world models and latent dynamics. A world model encodes an observation o_t into a latent state z_t and predicts its evolution under an action a_t, $z_{t+1} \approx F(z_t, a_t)$, enabling prediction and planning without repeatedly simulating high-dimensional observations. Modern latent world models typically learn the encoder and transition function jointly, supervising the dynamics through reconstructed observations or task-relevant latent targets (Ha & Schmidhuber, 2018; Hafner et al., 2020; Schrittwieser et al., 2020; Hansen et al., 2024). Joint-embedding predictive architectures

similarly learn representations by predicting in latent space rather than reconstructing every input detail (LeCun, 2022; Assran et al., 2023). Although effective, these approaches leave the geometry and transition operators of the latent space largely task-dependent.

Encoder outputs generally occupy a restricted manifold $\mathcal{M} = E(\mathcal{O})$ within the ambient latent space. Small one-step prediction errors can move autoregressive rollouts away from $\mathcal{M}$, causing errors to compound in regions where the transition model and decoder are poorly constrained. Latent-consistency objectives, contrastive learning, and architectural factorization can mitigate this drift (Kipf et al., 2020), but both the representation and its dynamics are generally learned anew for each observation domain.

Cognitive maps as structured dynamics. The cognitive-map hypothesis proposes that agents represent relations among states in a map-like form that supports flexible navigation and inference (Tolman, 1948). Place and grid cells provide complementary neural codes for location and path integration (O'Keefe & Dostrovsky, 1971; Hafting et al., 2005), while hippocampal–entorhinal activity also reflects abstract relational, conceptual, and task spaces (Constantinescu et al., 2016; Garvert et al., 2017). Predictive-map theories further relate hippocampal representations to expected future-state occupancy, with multiscale grid codes providing a compact basis for prediction and planning (Dayan, 1993; Stachenfeld et al., 2017; Momennejad et al., 2017).

These findings have motivated recurrent models in which path-integration and navigation objectives produce place- and grid-like representations (Banino et al., 2018; Cueva & Wei, 2018; Sorscher et al., 2019). The Tolman–Eichenbaum Machine separates transferable structural codes from sensory content (Whittington et al., 2020), while Vector-HaSH combines a pre-structured grid-cell scaffold with content storage, using low-dimensional shifts for high-capacity associative and episodic memory (Chandra et al., 2025). Hippocampal replay has likewise been linked to prospective evaluation and prioritized planning (Mattar & Daw, 2018), and structured activity slots model controllable recurrent subspaces as dynamic cognitive maps for sequence memory (Whittington et al., 2025). Together, these studies motivate structured latent dynamics as a reusable substrate for representation, prediction, and control.

Structural binding and compositional generalization. Structured world models aim to represent components and relations that can be recombined beyond the training distribution. Object-centric methods bind perceptual features into exchangeable slots (Locatello et al., 2020), whereas contrastively trained structured world models explicitly represent objects and their interactions (Kipf et al., 2020). Related approaches pursue disentangled factors of variation (Higgins et al., 2017; Locatello et al., 2019) or relational inductive biases (Battaglia et al., 2018) to promote systematic generalization. These forms of factorization can generalize to novel configurations, although they commonly prescribe the representational units and learn their transition rules jointly with downstream observations.

3 Pretrained Recurrent Modules Decompose Dynamics into a Block-Diagonal Scaffold

We first ask whether recurrent dynamics can be pretrained independently of downstream observations, and what geometric structure this imposes on the transition map. We train a bank of recurrent modules on two-dimensional path integration and freeze their dynamics. This section establishes two properties that the rest of the paper relies on: modules at different periods jointly provide stable, multiscale, and disambiguated spatial coordinates that increase representational bandwidth; and their independence makes the full transition Jacobian block diagonal, so that high-dimensional perturbations and actions decompose into per-module subspaces.

Each module is assigned a spatial period λ_m and represents position using two circular phases,

$$\boldsymbol{\phi}_m(x, y) = \left[\cos\left(\frac{2\pi x}{\lambda_m}\right), \sin\left(\frac{2\pi x}{\lambda_m}\right), \cos\left(\frac{2\pi y}{\lambda_m}\right), \sin\left(\frac{2\pi y}{\lambda_m}\right)\right]. \tag{1}$$

Each module is initialized from the phase of the starting position and is then driven only by two-dimensional displacement; it is trained to predict the phase at future positions, and absolute coordinates are never supplied as inputs or targets. We sample 20-step trajectories from random starts in a $2.2\,\mathrm{m} \times 2.2\,\mathrm{m}$ arena, supervise position and phase readouts with a mean-squared-error and a phase loss, and train eight bias-free modules of 512 ReLU units at distinct spatial periods for $100{,}000$ steps of 200 trajectories each.

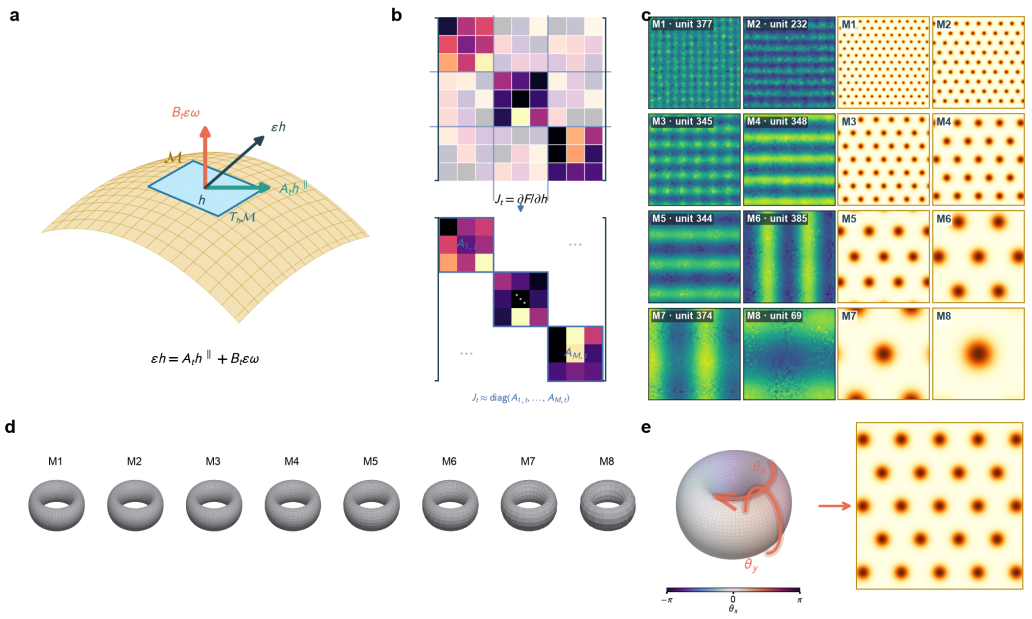


Figure 2: Geometry and modular structure of the pretrained recurrent scaffold. **(a)** A hidden-state perturbation decomposes into components tangent and normal to the learned manifold. **(b)** Independent recurrent modules constrain the transition Jacobian from a dense map to a block-diagonal operator. **(c)** Representative learned activity maps from modules M1–M8 (left) and their corresponding multiscale periodic spatial codes (right). **(d)** Each module represents two circular phases on a toroidal manifold, with spatial resolution varying across modules. **(e)** The two toroidal phase coordinates, θ_x and θ_y, generate a periodic two-dimensional spatial code.

For a fixed period, the two sine–cosine pairs define a coordinate on $S^1 \times S^1$, forming a two-torus for each module (Figure 2d–e). After pretraining, this toroidal coordinate is encoded in the recurrent hidden state and updated from self-motion. Representative hidden units exhibit repeated spatial fields at their modules' characteristic scales, including lattice-like, horizontal, and vertical response patterns (Figure 2c). These responses are termed *grid-cell-like* only in the limited sense of multiscale periodic coding; the separable phase target does not impose a hexagonal lattice(Burak & Fiete, 2009).

Any single periodic module aliases positions separated by integer multiples of its period, whereas coarse-period modules identify the broader spatial branch. Combining hidden states across periods preserves fine spatial detail while expanding the range over which positions can be distinguished. Per-module errors are approximately orthogonal and can mutually correct one another, while modules with distinct, ideally coprime, periods jointly index a much larger set of distinguishable spatial bins. The concatenated representation therefore increases representational bandwidth while the underlying physical variable remains two-dimensional. A post-hoc decoder recovers absolute position from the frozen states over 300-step open-loop rollouts, and comparisons between single modules and their concatenation confirm the multiscale disambiguation mechanism (Appendix A.2, A.3).

More importantly, the independence of the modules structures the dynamics themselves. Write the frozen hidden state as the direct sum of module states, $\mathbf{h}_t = \bigoplus_{m=1}^{M} \mathbf{h}_t^{(m)}$, and let F denote one recurrent step of the full scaffold. As illustrated in Figure 2a, a perturbation $\delta\mathbf{h}_t$—arising from an encoding error, an injected action, or off-manifold drift—can be decomposed into tangent and normal components and evolves to first order as

$$\delta\mathbf{h}_{t+1} = J_t\,\delta\mathbf{h}_t + \mathcal{O}(\|\delta\mathbf{h}_t\|^2), \qquad J_t = \left.\frac{\partial F}{\partial \mathbf{h}}\right|_{\mathbf{h}_t}. \tag{2}$$

Because module m evolves only from its own state and the shared displacement input, $\partial F^{(m)}/\partial \mathbf{h}^{(m')} = \mathbf{0}$ for $m \neq m'$. The full Jacobian is therefore block diagonal, as illustrated

in Figure 2b,

$$J_t = \bigoplus_{m=1}^{M} J_t^{(m)}, \qquad J_t^{(m)} = \left.\frac{\partial F^{(m)}}{\partial \mathbf{h}^{(m)}}\right|_{\mathbf{h}_t^{(m)}} \in \mathbb{R}^{d\times d}. \tag{3}$$

This structure reduces the effective transition parameters from $\mathcal{O}(M^2d^2)$ for a dense recurrent map to $\mathcal{O}(Md^2)$. Decomposing $\delta\mathbf{h}_t$ into a tangent component $\delta\mathbf{h}_t^{\parallel}$ along the learned manifold $\mathcal{M}$ and a normal component $\delta\mathbf{h}_t^{\perp}$ that drifts away from it, block-diagonality confines both components to within-module subspaces. Thus, single-step errors and injected controls cannot propagate directly across modules (Appendix A.1, Figure 5). This decomposition is the structural prior exploited by the downstream interface: a high-dimensional observation or action need only be resolved into a small set of module-specific two-dimensional controls.

These analyses establish that the pretrained module bank supplies a reusable latent coordinate system with known geometry, controllable dynamics, and a block-diagonal transition. The next section asks whether a learned router can bind distributed observations to this frozen scaffold and thereby recover the structure of a downstream task space.

4 Router Scheduling Recovers Task-Space Structure

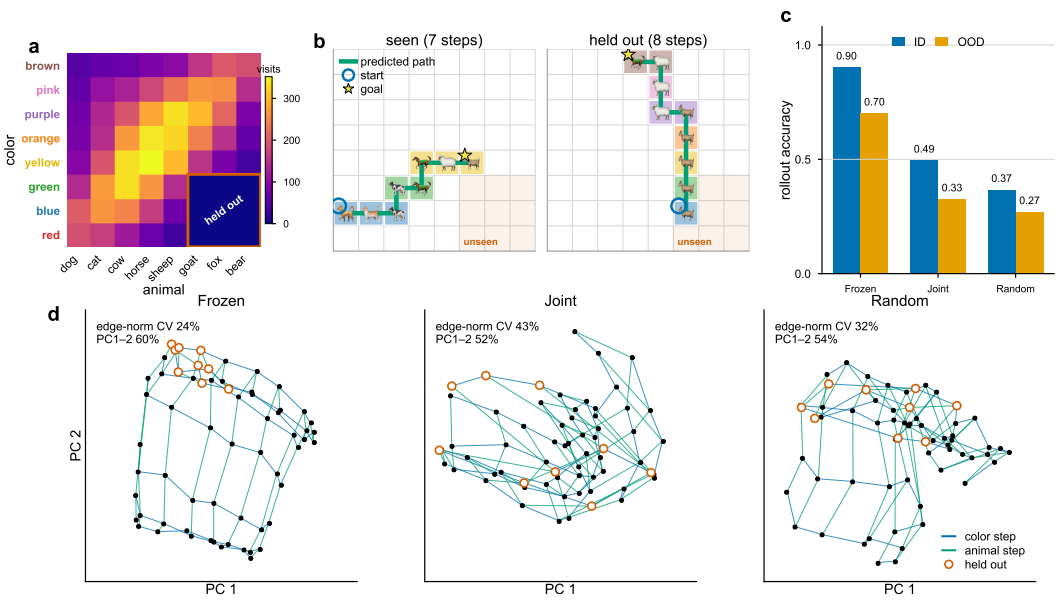


Figure 3: Few-shot planning emerges from a structured scaffold. **(a)** Training visits in the animal×color world; the outlined region is held out. **(b)** After limited training, the frozen scaffold predicts successful multistep paths for seen and held-out routes. **(c)** It achieves substantially higher ID and OOD rollout accuracy than joint and random baselines. **(d)** Its latent states recover a regular task-space topology, indicating that structured observation binding enables sample-efficient compositional generalization and planning.

Having fixed the recurrent dynamics, we ask whether a lightweight interface can rapidly bind a new observation domain to this pretrained coordinate system. A modality-specific encoder first produces a distributed representation of the current observation. A shared router then maps this representation, together with learned action embeddings, onto module-specific phase states and two-dimensional controls. Only the encoder, router, action interface, and decoder are optimized downstream; the scaffold itself remains frozen. The model receives no factor labels, coordinates, absolute state identifiers, or manually specified action directions.

Figure 3 demonstrates how this fixed scaffold supports few-shot learning in the controlled animal×color world. Training experience is limited and highly non-uniform, with an entire block of factor combinations excluded from training (Figure 3a). Nevertheless, the frozen model composes the learned animal and color transitions to produce successful multistep paths for both seen and held-out routes (Figure 3b). It consequently achieves substantially higher ID and OOD rollout

accuracy than both the jointly trained and random-scaffold baselines (Figure 3c). Because these alternatives use the same observation interface but must learn their transition geometry downstream, the comparison isolates the benefit of the pretrained scaffold rather than additional model capacity.

The latent geometry explains this sample efficiency. As shown in Figure 3d, the frozen scaffold organizes animal–color states into a regular graph with consistent transition lengths, while placing held-out combinations at coherent locations within the same structure. Jointly trained and random dynamics instead produce distorted and less systematic geometries (Whittington et al., 2025). The router therefore needs only to bind external observations to an existing structured z-space, rather than learn that space and its dynamics from scratch. Once this binding is established, the model can exploit the recovered task geometry to compose unseen states and plan routes from limited downstream experience.

5 Decomposition and Reassembly Yield Compositional Generalization

Because the router decomposes each observation into module-specific controls, previously learned factor values can be recombined without modifying the frozen dynamics. We evaluate this ability in factorized worlds whose training support contains every individual factor value and action but excludes selected combinations. Reconstruction, one-step prediction, and open-loop rollout are evaluated on seen states, unseen combinations, and trajectories initialized from held-out states (Figure 3a–c).

The language results in Table 1 clarify how task complexity affects this generalization. In the two-dimensional world, both modular models approach ceiling performance on held-out combinations, whereas the random reservoir fails even on the training distribution. The harder three-dimensional setting separates the models more clearly: the frozen scaffold reaches 50.5% open-loop accuracy, compared with 27.4% for joint training and 24.0% for the random baseline. Together with the successful held-out routes and regular latent geometry in Figure 3b–d, these results indicate that compositional generalization depends on organizing familiar factors within a reusable transition structure, rather than merely exposing the model to each factor individually.

Table 1: Master comparison across task families (rows = models, columns = tasks). Language worlds report open-loop rollout accuracy (OL, %) and closed-loop planning success (Plan, %); 3D Shapes reports rollout PSNR (dB) and goal-reaching planning success on seen (Plan$*$ID) and held-out (Plan$*$OOD) states (%); MetaWorld reports rollout MSE for ID/OOD splits as mean per-step (Roll) and final-step (Final) error, mean $\pm$ std over 10 seeds. Language 2D reports open-loop task-prediction accuracy on seen (ID) and held-out (OOD) factor combinations, mean $\pm$ std over 10 seeds. All results use the lightweight 20-epoch downstream training schedule; extended-training results are reported separately. Frozen (ours) row shaded.

	Language 2D		Language 3D		3D Shapes (pixel)			MetaWorld (state, MSE, ↓)			
Model	Acc$*$ID	Acc$*$OOD	OL	Plan	PSNR↑	Plan$*$ID ↑	Plan$*$OOD ↑	Roll$*$ID	Final$*$ID	Roll$*$OOD	Final$*$OOD
Frozen (ours)	$\mathbf{100.0 \pm 0.0}$	$\mathbf{99.8 \pm 0.1}$	$\mathbf{50.5 \pm 3.5}$	$\mathbf{36.0 \pm 1.2}$	$\mathbf{26.6}^{\dagger}$	**54.5**	**48.8**	$\mathbf{0.023 \pm 0.003}$	$\mathbf{0.040 \pm 0.006}$	$\mathbf{0.042 \pm 0.004}$	$\mathbf{0.076 \pm 0.007}$
Joint (modular)	100.0 ± 0.0	99.5 ± 0.3	27.4 ± 0.4	34.3 ± 1.0	$23.8^{\dagger}$	50.8	45.2	0.191 ± 0.006	0.256 ± 0.008	0.362 ± 0.012	0.488 ± 0.008
Random	51.7 ± 1.2	43.5 ± 0.9	24.0 ± 0.6	33.1 ± 1.1	11.0	43.9	36.1	0.269 ± 0.006	0.280 ± 0.009	0.489 ± 0.007	0.531 ± 0.008

The same decomposition extends from symbolic inputs to high-dimensional pixels. For DeepMind 3D Shapes (Burgess & Kim, 2018), convolutional encoders and decoders replace the language interface while the router, action interface, and recurrent scaffold remain unchanged. Factor annotations are used only to define evaluation splits and analyze the learned representation. Under the lightweight training schedule, the frozen scaffold achieves the highest aggregate rollout PSNR and preserves object identity, color, and scene structure over long horizons, whereas joint predictions progressively blur and drift (Table 1; Figure 4a). Its advantage on both ID and OOD planning further shows that the recovered factor structure remains usable for goal-directed prediction.

MetaWorld provides a more demanding test because its transitions involve continuous observations, continuous actions, and contact dynamics (Yu et al., 2020). The frozen scaffold nevertheless produces substantially lower rollout and endpoint errors on both ID and OOD splits (Table 1). The rendered trajectory in Figure 4b illustrates this difference: the scaffold prediction closely follows the reference motion and reaches the goal, while the jointly trained model accumulates larger hand-position errors. The consistent ordering across language, pixels, and embodied control suggests that decomposition and reassembly are properties of the shared scaffold rather than artifacts of a particular observation modality.

6 The Learned Codebook Binds Task-Relevant Dimensions and Enables Robust Planning

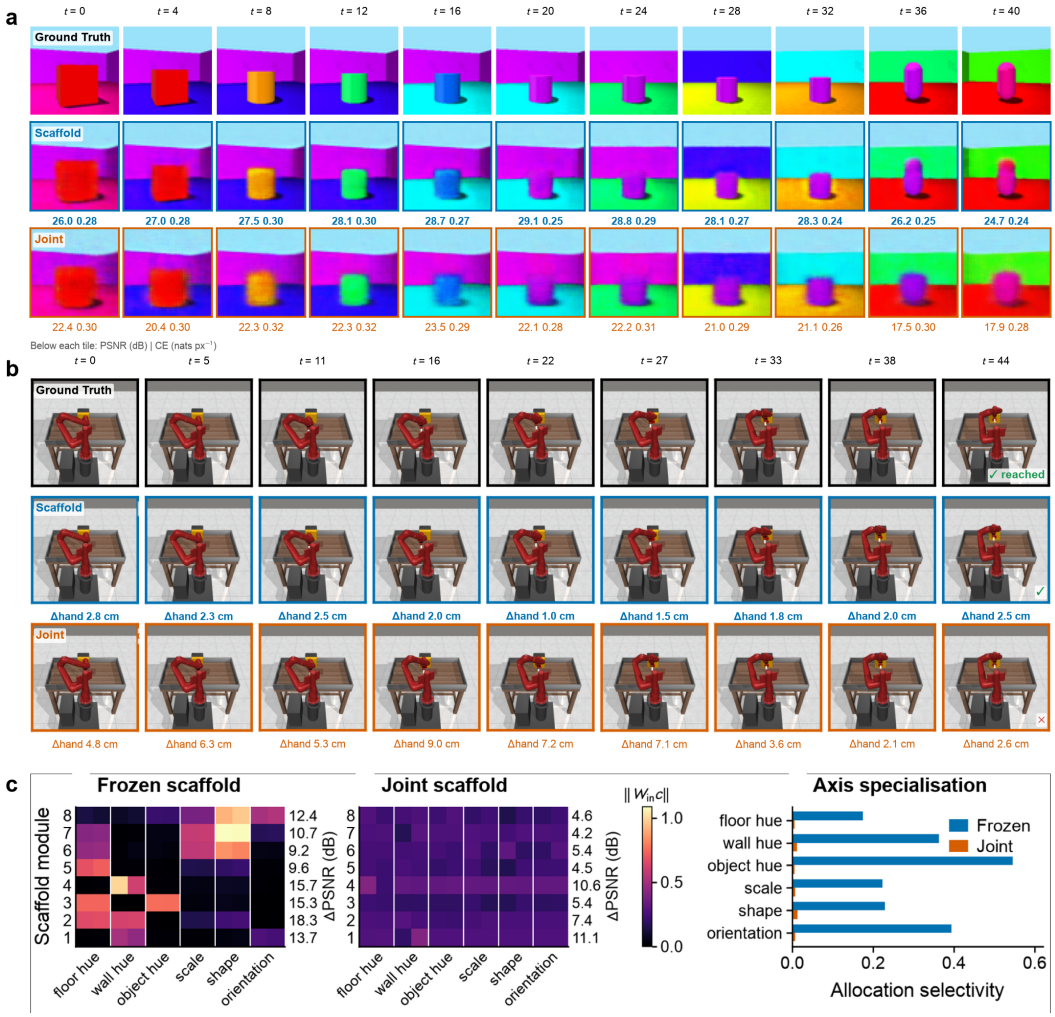


Figure 4: The pretrained scaffold supports accurate prediction and modular specialization across complex domains. **(a)** DeepMind 3D Shapes rollouts under the lightweight 20-epoch schedule; PSNR and cross-entropy are shown below each prediction. **(b)** Rendered MetaWorld trajectories with hand-position error. The frozen scaffold closely tracks the reference trajectory and reaches the goal, whereas the joint model accumulates larger errors. **(c)** Module–factor allocation and axis selectivity. Frozen modules specialize in distinct task dimensions, producing sparse, factor-specific assignments, while joint training yields diffuse allocation with little specialization. Numbers beside the heatmaps report the PSNR drop caused by ablating each module.

We next examine what the router binds to the frozen dynamics. The module–factor control norms in Figure 4c reveal a sparse allocation in which different scaffold modules respond preferentially to floor hue, wall hue, object hue, scale, shape, or orientation. The associated ablation scores show that these modules contribute materially to visual prediction. Joint training instead distributes most factors weakly across many modules and exhibits substantially lower axis selectivity.

This specialization provides a functional decomposition of the task space. Because the recurrent Jacobian is block diagonal, assigning different factors to different modules limits interference between their transitions: changing object hue need not alter the subspace representing shape or orientation. This organization resembles both structured activity slots (Whittington et al., 2025) and a mixture of experts (Shazeer et al., 2017; Fedus et al., 2022), but the experts here are recurrent dynamical subspaces rather than independently trained networks. Freezing the pretrained scaffold encourages

the router to reuse these subspaces, whereas joint training can reshape the dynamics and collapse factor-specific roles into a diffuse code.

The factor-aligned representation also supports planning. A goal-conditioned planner encodes the current and target observations through the same interface, selects actions from their latent-state difference, and rolls the scaffold forward without re-observing the environment (Racanière et al., 2017). Under the lightweight 20-epoch schedule summarized in Table 1, the frozen model achieves 26.6 dB rollout PSNR and reaches 54.5% and 48.8% planning success on ID and OOD states, respectively. With extended training—500 epochs of world-model-and-planner training with the recurrent scaffold kept frozen, followed by 100 epochs of planner-only refinement—the same model reaches 40.3 dB rollout PSNR and 77.6% route-planning accuracy. The stable visual predictions in Figure 4a indicate why additional planner optimization remains effective: the planner operates on a structured latent state that preserves task-relevant factors over long rollouts. Detailed performance across rollout horizons and goal distances is reported in Appendix A.4.

The MetaWorld trajectory in Figure 4b extends this connection to continuous control. The frozen scaffold tracks the reference hand motion with small deviations and reaches the target, while the joint model drifts and fails in the illustrated rollout. Quantitatively, its ID and OOD rollout MSEs are 0.023 and 0.042, substantially below the corresponding joint-model errors of 0.191 and 0.362 (Table 1). Module specialization is therefore not only an interpretability property: it supplies a structured coordinate system that can be exploited for prediction and planning. The remaining errors on contact-rich transitions also mark the boundary of a fixed toroidal scaffold and motivate extensions with local or hierarchical dynamics.

7 Conclusions

We introduced a hippocampal-inspired world model that binds observations and actions to a bank of pretrained, frozen recurrent modules. The modules supply a multiscale toroidal coordinate system with stable path integration; a learned router and action interface reuse it for compositional prediction and planning in language worlds; and the same frozen scaffold scales to pixel observations, where it learns quickly, drifts less, and recovers the topology of the task space. Across settings a frozen scaffold matches or outperforms jointly trained and random-reservoir baselines, indicating that reusable structure—not additional capacity—drives the gains.

8 Limitations and Future Works

The pretrained scaffold encodes translational, product-structured dynamics, and a fixed toroidal map need not suffice for strongly nonlinear or contact-rich transitions such as those in MetaWorld; characterizing where local or hierarchical extensions are required is an important direction. Imposing exact group-law constraints is also not always beneficial, suggesting that some slack in the structural prior can help. Performance still depends on the encoder, decoder, and planner; stronger perception, better planning, curricula, and comparisons to alternative pretrained world models (e.g. DINO-based (Caron et al., 2021; Oquab et al., 2024)) are natural next steps.

9 Reproducibility Statement

We describe the navigation environment, trajectory statistics, phase targets, recurrent architecture, and optimization settings in the supplementary material, including the exact recurrent update equations and the module periods. All downstream experiments reuse the same frozen scaffold, and the evaluation splits that define seen states, held-out compositions, and held-out starts are specified per experiment. Post-hoc decoders and probes used for analysis are trained on disjoint trajectories and never modify the frozen dynamics.

AI use statement

In this work, we used generative AI tools for language editing and for drafting and refactoring LaTeX. We have not used generative AI tools to generate experimental results, data, or figures, and generation of novel scientific claims was not delegated to such tools. We have reviewed all AI-assisted work: AI-edited text was checked by the authors for accuracy, and any AI-assisted code was verified and tested for correctness. We take responsibility for the final content of this work, including all text, claims, and artifacts.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023.

Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J. Chadwick, Thomas Degris, Joseph Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, 2018.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Timothy E. J. Behrens, Timothy H. Muller, James C. R. Whittington, Shirley Mark, Alon B. Baram, Kimberly L. Stachenfeld, and Zeb Kurth-Nelson. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509, 2018.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 1877–1901, 2020.

Yoram Burak and Ila R. Fiete. Accurate path integration in continuous attractor network models of grid cells. *PLoS Computational Biology*, 5(2):e1000291, 2009.

Chris Burgess and Hyunjik Kim. 3d shapes dataset. `https://github.com/google-deepmind/3d-shapes`, 2018.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

Sarthak Chandra, Sugandha Sharma, Rishidev Chaudhuri, and Ila Fiete. Episodic and associative memory from spatial scaffolds in the hippocampus. *Nature*, 638:739–751, 2025. doi: 10.1038/s41586-024-08392-y. URL `https://doi.org/10.1038/s41586-024-08392-y`.

Alexandra O. Constantinescu, Jill X. O'Reilly, and Timothy E. J. Behrens. Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292):1464–1468, 2016.

Christopher J. Cueva and Xue-Xin Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. In *International Conference on Learning Representations (ICLR)*, 2018.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Mona M. Garvert, Raymond J. Dolan, and Timothy E. J. Behrens. A map of abstract relational knowledge in the human hippocampal–entorhinal cortex. *eLife*, 6:e17086, 2017.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

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

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

Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*, 2017.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations (ICLR)*, 2020.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning (ICML)*, pp. 2873–2882, 2018.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview, June 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf. Version 0.9.2, position paper.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning (ICML)*, pp. 4114–4124, 2019.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 11525–11538, 2020.

Marcelo G. Mattar and Nathaniel D. Daw. Prioritized memory access explains planning and hippocampal replay. *Nature Neuroscience*, 21(11):1609–1617, 2018.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Ida Momennejad, Evan M. Russek, Jin H. Cheong, Matthew M. Botvinick, Nathaniel D. Daw, and Samuel J. Gershman. The successor representation in human reinforcement learning. *Nature Human Behaviour*, 1(9):680–692, 2017.

John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175, 1971.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 27730–27744, 2022.

Sébastien Racanière, Théophane Weber, David Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adrià Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

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, 2020.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Ben Sorscher, Gabriel C. Mel, Surya Ganguli, and Samuel A. Ocko. A unified theory for the origin of grid cells through the lens of pattern formation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Kimberly L. Stachenfeld, Matthew M. Botvinick, and Samuel J. Gershman. The hippocampus as a predictive map. *Nature Neuroscience*, 20(11):1643–1653, 2017.

Erik Talvitie. Self-correcting models for model-based reinforcement learning. In *AAAI Conference on Artificial Intelligence*, volume 31, 2017.

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

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

James C. R. Whittington, Timothy H. Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy E. J. Behrens. The tolman–eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5):1249–1263.e23, 2020. doi: 10.1016/j.cell.2020.10.024. URL `https://doi.org/10.1016/j.cell.2020.10.024`.

James C. R. Whittington, David McCaffary, Jacob J. W. Bakermans, and Timothy E. J. Behrens. How to build a cognitive map. *Nature Neuroscience*, 25(10):1257–1272, 2022.

James C. R. Whittington, William Dorrell, Timothy E. J. Behrens, Surya Ganguli, and Mohamady El-Gaby. A tale of two algorithms: Structured slots explain prefrontal sequence memory and are unified with hippocampal cognitive maps. *Neuron*, 113(2):321–333.e6, 2025. doi: 10.1016/j.neuron.2024.10.017. URL `https://doi.org/10.1016/j.neuron.2024.10.017`.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, pp. 1094–1100, 2020.

A SUPPLEMENTARY INFORMATION

A.1 PERTURBATION THEORY AND MODULAR JACOBIAN STRUCTURE

Consider a discrete nonlinear dynamical system

$$\mathbf{h}_{t+1} = F(\mathbf{h}_t, \mathbf{u}_t), \tag{4}$$

where $\mathbf{h}_t \in \mathbb{R}^H$ is the hidden state and $\mathbf{u}_t$ is the control input. Let $\widetilde{\mathbf{h}}_t = \mathbf{h}_t + \delta\mathbf{h}_t$ denote a nearby perturbed state. Holding the control fixed and applying a first-order Taylor expansion around $\mathbf{h}_t$ gives

$$F(\mathbf{h}_t + \delta\mathbf{h}_t, \mathbf{u}_t) = F(\mathbf{h}_t, \mathbf{u}_t) + J_t\delta\mathbf{h}_t + \mathcal{O}\big(\|\delta\mathbf{h}_t\|^2\big), \tag{5}$$

$$J_t = \left.\frac{\partial F(\mathbf{h}, \mathbf{u}_t)}{\partial \mathbf{h}}\right|_{\mathbf{h}=\mathbf{h}_t}. \tag{6}$$

Subtracting the unperturbed transition yields the first-order perturbation dynamics

$$\delta\mathbf{h}_{t+1} = J_t\delta\mathbf{h}_t + \mathcal{O}\big(\|\delta\mathbf{h}_t\|^2\big). \tag{7}$$

Thus, the Jacobian evaluated at the perturbed operating point provides the local linear map that predicts how a small state error evolves over one recurrent step.

A simple two-dimensional example makes this approximation explicit. Consider

$$F(x, y) = \begin{bmatrix} x + \alpha \sin y \\ y + \beta x^2 \end{bmatrix}. \tag{8}$$

Its Jacobian is

$$J(x, y) = \begin{bmatrix} 1 & \alpha \cos y \\ 2\beta x & 1 \end{bmatrix}. \tag{9}$$

At the operating point $\mathbf{h}_* = (x_*, y_*)^\top$, a perturbation $\delta\mathbf{h} = (\delta x, \delta y)^\top$ therefore evolves as

$$F(\mathbf{h}_* + \delta\mathbf{h}) - F(\mathbf{h}_*) \approx J(x_*, y_*)\delta\mathbf{h} = \begin{bmatrix} \delta x + \alpha \cos(y_*)\delta y \\ 2\beta x_*\delta x + \delta y \end{bmatrix}. \tag{10}$$

The omitted terms, including $-\frac{\alpha}{2}\sin(y_*)\delta y^2$ and $\beta\delta x^2$, are second order in the perturbation. The approximation therefore becomes increasingly accurate as $\|\delta\mathbf{h}\|$ decreases.

For a state $\mathbf{h}_t$ lying on a learned manifold $\mathcal{M}$, the perturbation can be decomposed into tangent and normal components,

$$\delta\mathbf{h}_t = P_{T_t}\delta\mathbf{h}_t + P_{N_t}\delta\mathbf{h}_t \equiv \delta\mathbf{h}_t^{\parallel} + \delta\mathbf{h}_t^{\perp}, \tag{11}$$

where P_{T_t} and P_{N_t} project onto the tangent and normal spaces of $\mathcal{M}$ at $\mathbf{h}_t$. Equation 7 then gives

$$\delta\mathbf{h}_{t+1} \approx J_t\delta\mathbf{h}_t^{\parallel} + J_t\delta\mathbf{h}_t^{\perp}. \tag{12}$$

The tangent component changes the represented state along the learned manifold, whereas the normal component corresponds to off-manifold error (Figure 5a).

For M independent recurrent modules of width d, the full hidden state is $\mathbf{h}_t = \bigoplus_{m=1}^{M} \mathbf{h}_t^{(m)}$. Since module m depends only on its own state and the shared control,

$$\frac{\partial F^{(m)}}{\partial \mathbf{h}^{(m')}} = \mathbf{0}, \qquad m \neq m', \tag{13}$$

and the Jacobian becomes

$$J_t = \operatorname{diag}\left(J_t^{(1)}, J_t^{(2)}, \ldots, J_t^{(M)}\right). \tag{14}$$

A dense recurrent transition can couple all $H = Md$ hidden dimensions (Figure 5b), whereas the modular scaffold admits only within-module first-order interactions (Figure 5c). This reduces the number of potentially nonzero Jacobian entries from $\mathcal{O}(M^2d^2)$ to $\mathcal{O}(Md^2)$ and prevents a perturbation in one module from immediately propagating into every other module.

A.2 OPEN-LOOP PATH-INTEGRATION EVALUATION

We roll out the frozen scaffold from the initial phase using displacement alone over 300 steps, which is $15\times$ the 20-step training horizon. Figure 6 shows six representative trajectories. The decoded paths closely follow the ground-truth trajectories, with median errors of 0.7–3.2 cm. One

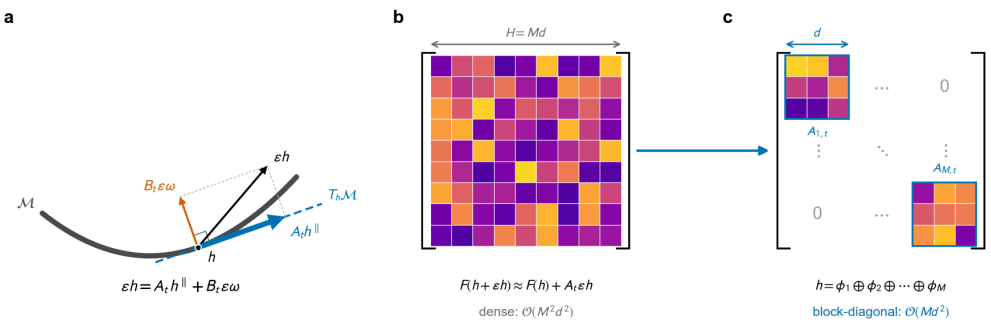


Figure 5: Perturbation decomposition and modular Jacobian structure. **(a)** A local perturbation decomposes into components tangent and normal to the learned manifold. **(b)** A dense Jacobian couples all $H = Md$ hidden dimensions. **(c)** Independent recurrent modules produce a block-diagonal Jacobian, restricting first-order perturbation propagation to within-module subspaces.

example exhibits a half-period drift in which the decoder settles on a neighbouring periodic location, illustrating the aliasing failure mode addressed by multiscale disambiguation.

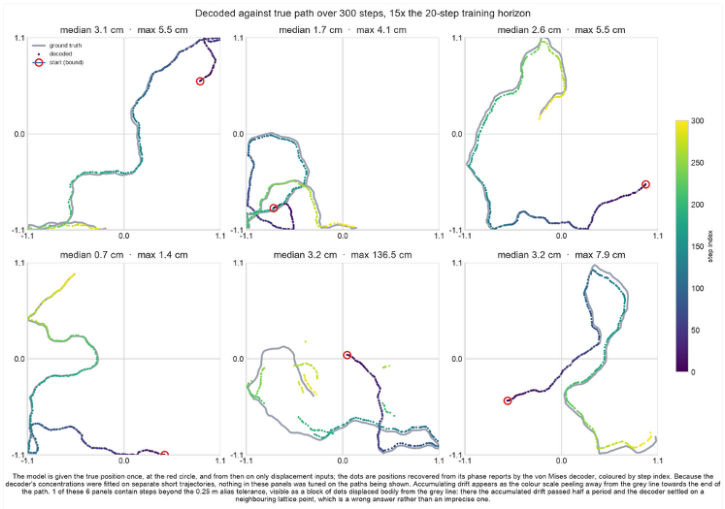


Figure 6: Decoded and ground-truth positions over six 300-step open-loop rollouts. Dot color indicates the rollout step; median and maximum position errors are annotated above each panel. Spatial separation between the decoded points and the grey ground-truth trajectory indicates accumulated prediction error.

A.3 Multiscale joint disambiguation

We formalize how modules with different spatial periods progressively eliminate the aliases produced by any individual periodic code. Consider first a one-dimensional interval of length L, discretized into $N = L/\Delta$ spatial bins of width Δ. Let module m have period $p_m = \lambda_m/\Delta$ bins and encode position n using the circular phase

$$\phi_m(n) = \left[\cos\left(\frac{2\pi n}{p_m}\right), \sin\left(\frac{2\pi n}{p_m}\right)\right]. \tag{15}$$

The full sine–cosine pair uniquely identifies phase within one period but repeats every p_m bins. Given the phase observed at a true position n_*, module m therefore admits the candidate set

$$\mathcal{C}_m(n_*) = \{n \in \{0, \ldots, N-1\} : n \equiv n_* \pmod{p_m}\}, \tag{16}$$

whose size is approximately

$$|\mathcal{C}_m(n_*)| \approx \frac{N}{p_m} = \frac{L}{\lambda_m}. \tag{17}$$

Thus, a single module with period λ_1 leaves approximately L/λ_1 indistinguishable positions along the interval.

Adding a second module requires a valid position to satisfy both periodic constraints,

$$n \equiv n_* \pmod{p_1}, \qquad n \equiv n_* \pmod{p_2}. \tag{18}$$

The joint candidate set is the intersection $\mathcal{C}_{1,2} = \mathcal{C}_1 \cap \mathcal{C}_2$. Its solutions repeat with joint period

$$P_{1,2} = \mathrm{lcm}(p_1, p_2), \tag{19}$$

and hence

$$|\mathcal{C}_{1,2}| \approx \frac{N}{\mathrm{lcm}(p_1, p_2)} = \frac{N \ \gcd(p_1, p_2)}{p_1 p_2}. \tag{20}$$

If $p_2 = p_1$, the second module provides no additional disambiguation. If the periods differ but share factors, the candidate set shrinks according to their greatest common divisor. For coprime periods, $\gcd(p_1, p_2) = 1$, Equation 20 reduces to

$$|\mathcal{C}_{1,2}| \approx \frac{N}{p_1 p_2}. \tag{21}$$

The second module therefore removes a fraction $1 - 1/p_2$ of the candidates retained by the first module. More generally, for M modules,

$$\left| \bigcap_{m=1}^{M} \mathcal{C}_m(n_*) \right| \approx \frac{N}{\mathrm{lcm}(p_1, p_2, \ldots, p_M)}. \tag{22}$$

When the periods are pairwise coprime, the joint repetition length grows as $\prod_m p_m$, so the number of aliases decreases multiplicatively. For incommensurate real-valued periods, exact repetitions need not occur within a finite interval; with finite decoding tolerance, the same principle produces a single dominant candidate once the joint period exceeds the represented range.

The two-dimensional scaffold applies this argument independently to the x- and y-phase pairs. Equivalently, if each module defines a likelihood $p(\mathbf{r}_m \mid \mathbf{x})$, the joint posterior is

$$p(\mathbf{x} \mid \mathbf{r}_1, \ldots, \mathbf{r}_M) \propto p(\mathbf{x}) \prod_{m=1}^{M} p(\mathbf{r}_m \mid \mathbf{x}). \tag{23}$$

Each additional module suppresses spatial locations inconsistent with its phase, progressively sharpening a periodic lattice of candidates into a single posterior mode.

Figure 7 quantifies this process. Panel (a) reports decoding error over rollout time; panel (b) compares single-module and joint position decoding; panel (c) compares decoded and true coordinates; panel (d) shows the error distribution and aliasing threshold; panel (e) maps spatial error; and panel (f) shows that both RMSE and alias rate decrease as modules are added.

Figure 8 visualizes the corresponding two-dimensional posterior. A single module produces a dense periodic lattice of candidate positions. Intersecting the constraints from two and four modules removes most aliases, while the full eight-module bank produces a single sharp mode at the true position.

A.4 Visual planning across goal distance and rollout horizon

Figure 9 compares frozen and jointly trained scaffolds on DeepMind 3D Shapes as a function of goal distance and rollout horizon. Panel (a) reports open-loop goal-reaching success, panel (b) tracks predicted frame cross-entropy over rollout time, and panel (c) evaluates the predicted stop frame as a function of goal distance. The frozen scaffold retains high goal-reaching accuracy and lower prediction error across long distances, whereas the joint model fails to reach most goals and exhibits unstable stop-frame predictions.

A.5 Recovered task-space topology in visual worlds

Figure 10 quantifies how faithfully the frozen scaffold organizes visual observations within its pretrained coordinate system, supporting the structure and specialization results in Sections 5 and 6. Along the scale axis, the frozen representation produces nearly uniform latent transitions. The same regularity extends to the two-dimensional scale–floor-hue subspace and the three-dimensional hue subspace. A nearest-neighbour analysis in the full 4096-dimensional hidden space further tests whether local task-space adjacency is preserved without relying on a low-dimensional projection.

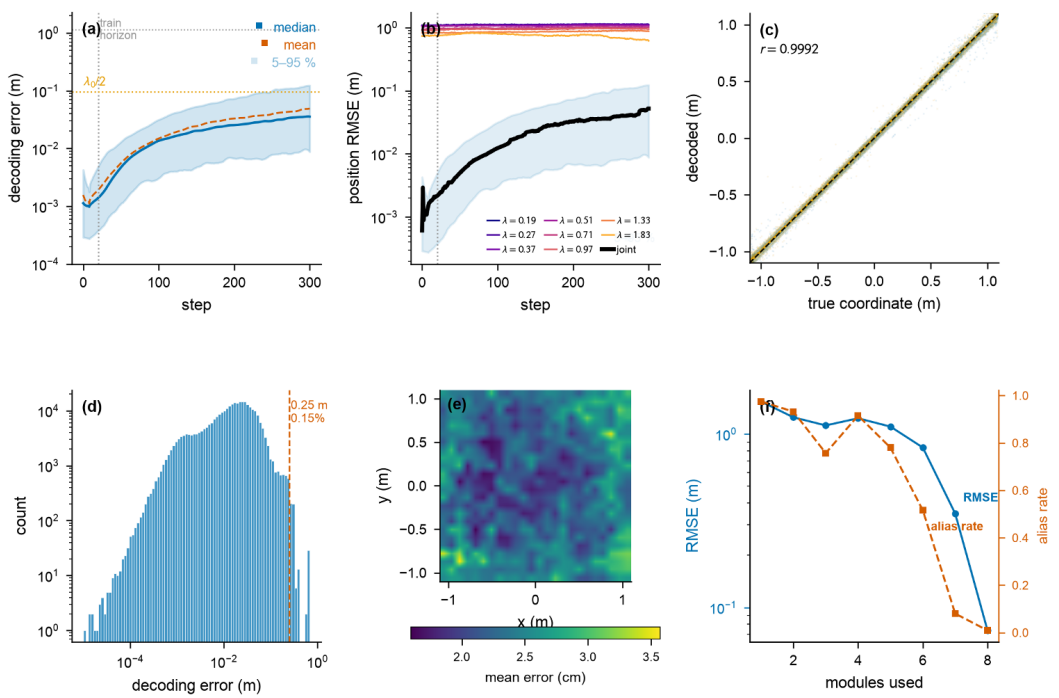


Figure 7: Path-integration and multiscale-disambiguation diagnostics. **(a)** Decoding error over roll-out steps. **(b)** Position RMSE for individual modules and their joint representation. **(c)** Decoded versus true coordinates. **(d)** Distribution of decoding errors; the dashed line marks the $0.25\,\mathrm{m}$ alias threshold. **(e)** Spatial distribution of mean error. **(f)** Joint RMSE and alias rate as the number of modules increases.

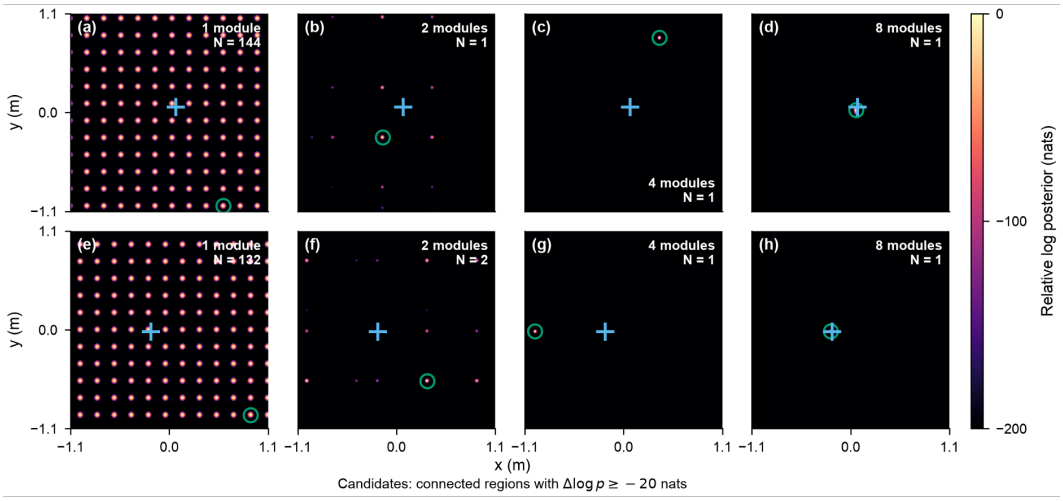


Figure 8: Joint posterior disambiguation for two example positions (rows). Columns use 1, 2, 4, and 8 modules. The cyan cross denotes the true position, the green circle denotes the decoded posterior mode, and N is the number of connected candidate regions satisfying $\Delta \log p \geq -20$ nats. Combining modules with distinct periods progressively reduces a periodic lattice of aliases to a single posterior mode.

A.6 Trajectory Generation

Scaffold-pretraining trajectories are sampled in a bounded $2.2\,\mathrm{m} \times 2.2\,\mathrm{m}$ environment. Each trajectory contains 20 transitions and begins from a randomly sampled position. The recurrent modules receive displacement $\Delta \mathbf{p}_t = \mathbf{v}_t \Delta t$ rather than absolute position. Training trajectories are sampled online, while independently seeded validation and test trajectories are held fixed. Open-loop evaluation extends each trajectory to 300 steps without providing corrective observations.

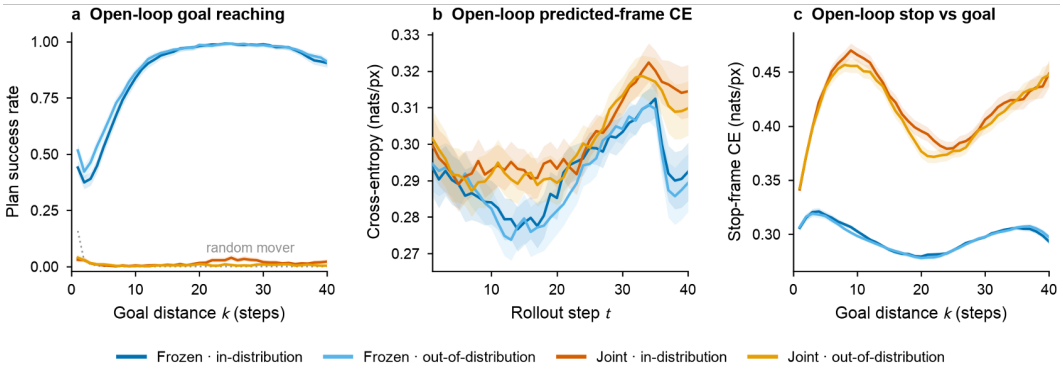


Figure 9: Frozen and jointly trained scaffolds on DeepMind 3D Shapes. **(a)** Open-loop planning success as a function of goal distance. **(b)** Predicted-frame cross-entropy over the rollout horizon. **(c)** Stop-frame cross-entropy as a function of goal distance. Blue denotes the frozen scaffold, orange denotes the joint baseline, and dark and light curves denote ID and OOD evaluation, respectively.

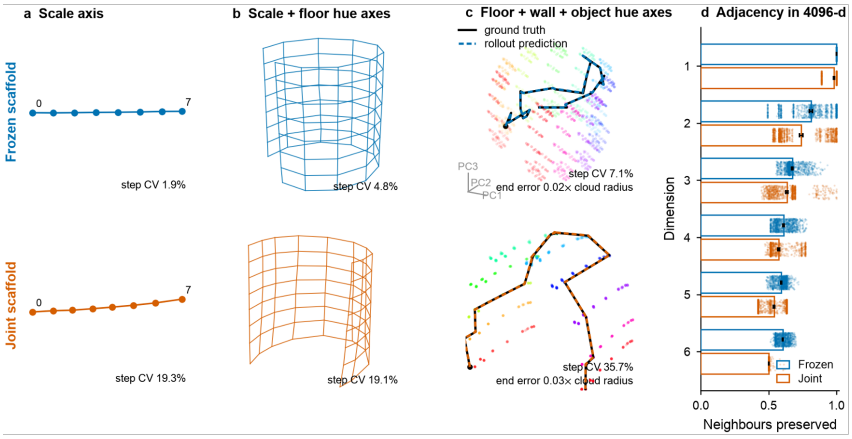


Figure 10: Recovered task-space topology. **(a)** Along the scale axis, the frozen scaffold produces evenly spaced latent steps, whereas the joint baseline is irregular. **(b)** In the scale–floor-hue subspace, the frozen representation forms a regular cylindrical grid while the joint representation is warped. **(c)** In the floor-, wall-, and object-hue subspace, the frozen rollout tracks the ground-truth trajectory with lower step variation and endpoint error. **(d)** Nearest-neighbour preservation in the full 4096-dimensional hidden space across all six task factors.

A.7 Periodic phase targets

Module m uses the phase target in Equation 1. The eight spatial periods are

$$\lambda_m \in \{0.19,\ 0.27,\ 0.37,\ 0.51,\ 0.71,\ 0.97,\ 1.33,\ 1.83\}\ \mathrm{m}. \tag{24}$$

Each module therefore represents a pair of circular phases and cannot by itself distinguish positions separated by integer multiples of λ_m. Combining the modules yields the joint disambiguation mechanism derived in Equations 20–22.

A.8 Recurrent architecture

Each module is an independent, bias-free recurrent network with hidden width $d = 512$, a ReLU nonlinearity, an initialization map, a recurrent transition, and a linear phase readout:

$$\mathbf{h}_0^{(m)} = \tanh\left(W_{\text{init}}^{(m)} \boldsymbol{\phi}_m(\mathbf{p}_0)\right), \tag{25}$$

$$\mathbf{h}_{t+1}^{(m)} = \mathrm{ReLU}\left(W_{\text{rec}}^{(m)} \mathbf{h}_t^{(m)} + W_{\text{in}}^{(m)} \Delta\mathbf{p}_t\right), \tag{26}$$

$$\widehat{\boldsymbol{\phi}}_{t+1}^{(m)} = W_{\text{phase}}^{(m)} \mathbf{h}_{t+1}^{(m)}. \tag{27}$$

Modules do not share parameters and do not receive module identities, time indices, or non-periodic position channels. Their hidden states are concatenated in order of spatial period. All recurrent parameters are frozen during downstream world-model and planning experiments.

A.9 PRETRAINING OBJECTIVE

The phase-prediction objective is

$$\mathcal{L}_{\text{phase}} = \frac{1}{BTM} \sum_{b=1}^{B} \sum_{t=1}^{T} \sum_{m=1}^{M} \left\| \widehat{\boldsymbol{\phi}}_{b,t}^{(m)} - \boldsymbol{\phi}_{b,t}^{(m)} \right\|_2^2 . \tag{28}$$

The loss is averaged over batches, rollout steps, and modules. Absolute coordinates, downstream factor labels, and downstream task supervision are not provided to the recurrent modules.

A.10 OPTIMIZATION AND COMPUTE

Table 2: Scaffold-pretraining hyperparameters.

Stage	Hyperparameter	Setting
Scaffold pretraining	Number of modules	8
Scaffold pretraining	Spatial periods λ_m (m)	0.19, 0.27, 0.37, 0.51, 0.71, 0.97, 1.33, 1.83
Scaffold pretraining	Hidden width per module	512
Scaffold pretraining	Activation and bias	ReLU; no bias
Scaffold pretraining	Arena size	$2.2\,\text{m} \times 2.2\,\text{m}$
Scaffold pretraining	Training horizon	20 steps
Scaffold pretraining	Batch size	200 trajectories
Scaffold pretraining	Training duration	100,000 optimization steps
Scaffold pretraining	Optimizer	*Adam*
Scaffold pretraining	Learning rate	*1e-3*