SPATIAL WORLD MODEL: SPATIALLY STRUCTURED LATENT DYNAMICS AND PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent world models have emerged as a compelling approach to learning environment dynamics through reward-free representation learning. However, their latent dynamics often remain entangled and geometrically unstructured, limiting their interpretability and utility for downstream planning. In this paper, we introduce *Spatial World Model* (SWM), a framework for learning spatially structured latent dynamics. SWM structures the latent space through two complementary geometric properties: orthogonality and directional ordering. Orthogonality enforces mutually orthogonal latent directions, while directional ordering arranges each latent-transition projection according to its corresponding state-transition component. These properties yield a spatially structured latent space that reflects the spatial organization of the environment. Building on this structure, we construct a goal-directed latent heuristic by projecting the current-to-goal latent displacement onto the learned directions. We incorporate this heuristic into *Spatial-Aware Latent Planning* (SLP), which guides search over a latent roadmap and composes local world-model predictions into long-horizon plans. Experiments on OGBench PointMaze validate the spatial structure learned by SWM and show that SLP outperforms comparison methods at larger goal offsets and longer execution horizons.

1 INTRODUCTION

World models (WMs) learn internal representations of an environment and predict how these representations evolve under actions. These predictions enable an agent to reason about potential future states. However, accurate prediction alone does not ensure that a representation provides clear geometric guidance for planning. When latent transitions are entangled and geometrically unstructured, a planner cannot directly infer how movement in latent space relates to spatial movement in the environment. This motivates organizing predictive latent dynamics so that their geometry also reflects spatial movement.

Joint-Embedding Predictive Architectures (JEPAs) (Assran et al., 2023) provide a natural foundation for predictive latent world modeling. JEPA-style world models predict future representations directly in latent space, avoiding pixel-level reconstruction and task-specific reward supervision. Recent approaches improve representation quality and predictive stability through distributional regularization (Maes et al., 2026), pretrained visual representations (Siméoni et al., 2025; Zhou et al., 2024), and other latent constraints (Sobal et al., 2025). Nevertheless, these objectives do not explicitly organize latent dynamics according to spatial movement in the environment. Accurate latent prediction may therefore coexist with a geometry that provides little directional guidance for long-horizon planning.

Inspired by evidence that the hippocampal–entorhinal system forms structured representations of locations and routes (Moser et al., 2008), we introduce the *Spatial World Model* (SWM). SWM uses spatial state transitions to organize latent-transition vectors along learned directions. It imposes two complementary geometric properties: *orthogonality* and *directional ordering*. SWM learns one latent direction for each spatial state-transition component and enforces mutual orthogonality among these directions, while directional ordering organizes the projected transitions according to the corresponding state-transition values. These properties produce a spatially structured latent space that reflects the spatial organization of the environment (Fig. 1).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

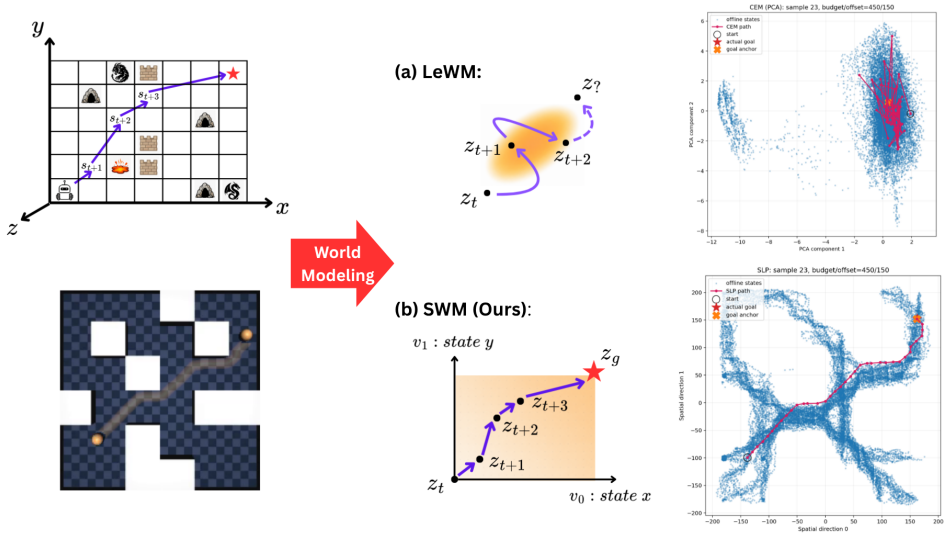


Figure 1: **Conceptual comparison and planning example for LeWM and SWM.** The left column illustrates spatial transitions and the PointMaze task used in the planning example. **(a)** LeWM learns predictive latent dynamics without explicitly organizing latent-transition geometry. The latent states provide little correspondence to the spatial structure of the environment. **(b)** SWM organizes latent-transition vectors through orthogonality and directional ordering. The learned spatial directions reflect the spatial organization of the environment and allow SLP to construct a goal-directed path.

This spatial structure also offers direct guidance for planning. Latent world-model planners commonly rely on sampling-based trajectory optimization, such as CEM (Rubinstein, 1997; Hafner et al., 2019) or MPPI (Hansen et al., 2022; 2024), to search over action sequences. While these methods can be effective, their search procedures operate without explicitly exploiting the spatial structure of learned latent transitions. In SWM, we construct a goal-directed latent heuristic by projecting the current-to-goal latent displacement onto the learned directions. We incorporate this heuristic into *Spatial-Aware Latent Planning* (SLP), which guides search over a latent roadmap and composes local world-model predictions into long-horizon plans.

We evaluate SWM on OGBench PointMaze *Medium*, *Large*, and *Giant* (Park et al., 2025). Our experiments examine the geometry of the learned latent dynamics, long-horizon planning performance, and landmark-retrieval efficiency. The results show that SWM learns orthogonal and directionally ordered latent dynamics. SLP outperforms comparison methods at larger goal offsets and longer horizons, while projected-space retrieval reduces landmark-matching cost without degrading planning performance.

Our contributions are summarized as follows:

- We introduce the *Spatial World Model*, which learns spatially structured latent dynamics that reflect the spatial organization of the environment through orthogonality and directional ordering.
- Building on this structure, we develop *Spatial-Aware Latent Planning*, which constructs a goal-directed heuristic, guides search over a latent roadmap, and composes local world-model predictions into long-horizon plans.
- We evaluate SWM and SLP on OGBench PointMaze, showing that SWM learns the intended spatial structure and that SLP outperforms comparison methods over longer planning horizons. We further show that kNN retrieval in the projected transition space accelerates landmark matching without degrading planning performance.

2 Related Work

Reward-Free Latent World Models. Classical world models combine latent dynamics with reconstruction, reward prediction, or control objectives (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2021; Schrittwieser et al., 2020; Hansen et al., 2022; 2024). JEPAs instead learn representations by predicting target embeddings without reconstructing observations (Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025). Reward-free latent world models extend this principle through pretrained visual features, distributional regularization, and auxiliary latent objectives (Zhou et al., 2024; Maes et al., 2026; Saanum et al., 2024). These approaches primarily emphasize predictive accuracy and representation quality, while the spatial organization of their latent dynamics remains implicit. SWM builds on this predictive paradigm. In addition to predicting future representations, it explicitly aligns latent geometry with spatial state structure.

Structured Latent Dynamics. Prior work structures latent dynamics through object factorization (Kipf et al., 2020; Nam et al., 2026), discrete representations (Micheli et al., 2023), or simplified action effects (Saanum et al., 2024). These formulations introduce useful structure for representing objects, actions, or discrete dynamics. SWM addresses a complementary aspect by organizing latent dynamics according to the spatial structure of the environment.

Planning in Learned Latent Spaces. Latent world models commonly optimize action sequences using CEM or MPPI (Rubinstein, 1997; Hafner et al., 2020; Zhou et al., 2024; Maes et al., 2026; Hansen et al., 2022; 2024). These sampling-based planners search directly over predicted rollouts without requiring an explicit global latent map. Other approaches introduce latent connectivity or geometry: LSR (Lippi et al., 2022) and L3P (Zhang et al., 2021) construct latent roadmaps, while TSLP (Wang et al., 2026b) straightens latent trajectories. AdaJEPA (Wang et al., 2026a) adapts the model during control, and RC-Aux (Li et al., 2026) learns budget-conditioned reachability. SWM connects these perspectives by learning spatial coordinates jointly with predictive dynamics and using them for transition matching, heuristic roadmap search, and efficient landmark retrieval.

3 Method

Spatial World Model (SWM) has three components. First, an action-conditioned latent world model encodes observations and predicts their future representations. Second, our *State-Transition-Ranked Orthogonal Latent Construction* organizes latent-transition vectors using spatial state transitions. It learns one latent direction for each spatial state-transition component, encourages mutual orthogonality among these directions, and orders projected transitions by the associated component values. Third, *Spatial-Aware Latent Planning* (SLP) uses the learned directions to connect local world-model predictions to offline latent landmarks and construct a goal-directed heuristic that guides long-horizon search over a latent roadmap.

3.1 Latent World Model

We build SWM on an action-conditioned latent world model with an observation encoder $Enc(\cdot)$ and a latent predictor $Pred(\cdot, \cdot)$. Given a transition (o_t, a_t, o_{t+1}), the encoder maps the current observation to a latent representation, while the next observation provides a stop-gradient prediction target:

$$z_t = Enc(o_t), \qquad z_{t+1} = \mathrm{sg}(Enc(o_{t+1}))\,, \tag{1}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation. The predictor takes the current latent representation z_t and action a_t and predicts the next latent representation:

$$\hat{z}_{t+1} = Pred(z_t, a_t). \tag{2}$$

We train the model using the latent prediction loss

$$\mathcal{L}_{\text{pred}} = \|\hat{z}_{t+1} - z_{t+1}\|_2^2\,. \tag{3}$$

SWM retains this prediction objective and additionally organizes how environment transitions are represented in latent space. Let $s_t \in \mathbb{R}^K$ denote the spatial state associated with observation o_t;

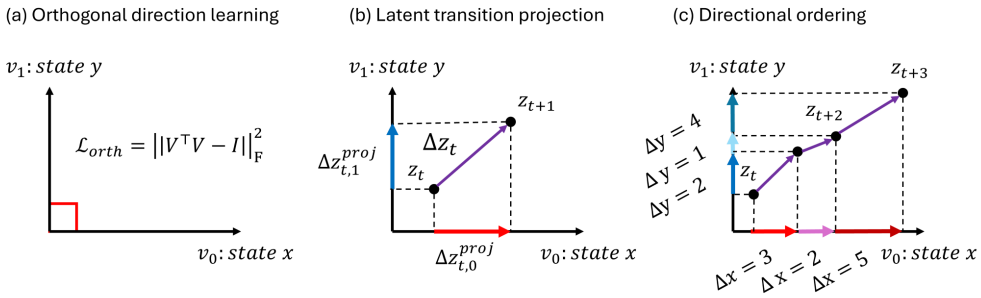


Figure 2: **State-transition-ranked orthogonal latent construction. (a)** SWM learns mutually orthogonal spatial directions. **(b)** Latent transitions (Δz_t) are projected onto these directions. **(c)** Each projected coefficient is ordered by its corresponding state-transition component.

for PointMaze, $s_t = (x_t, y_t)^\top$. We define the latent transition and its corresponding spatial state transition as

$$\Delta z_t = z_{t+1} - z_t, \qquad \Delta s_t = s_{t+1} - s_t. \tag{4}$$

The action a_t remains the input to the latent predictor, while Δs_t provides the spatial supervision used to organize Δz_t. SWM therefore imposes structure on latent-transition vectors rather than directly constraining individual latent states.

3.2 State-Transition-Ranked Orthogonal Latent Construction

Our central idea is to organize latent transitions according to their corresponding spatial state transitions. SWM learns one latent direction for each spatial state-transition component and encourages these directions to be mutually orthogonal. It then organizes the projected latent transitions along each direction according to the corresponding state-transition values. We achieve this through two complementary components: *orthogonal direction learning* and *directional ranking*.

Orthogonal Direction Learning. Let D denote the latent dimension and K the number of spatial state-transition components. We parameterize the learned latent directions as

$$V = [v_0, v_1, \ldots, v_{K-1}] \in \mathbb{R}^{D \times K}, \tag{5}$$

where $v_k \in \mathbb{R}^D$ is the latent direction associated with the k-th spatial state-transition component. We therefore refer to the learned directions $\{v_k\}_{k=0}^{K-1}$ as *spatial directions* throughout the paper. These directions are optimized jointly with the latent world model.

We encourage the learned spatial directions to be mutually orthogonal using

$$\mathcal{L}_{\text{orth}} = \left\| V^\top V - I_K \right\|_F^2, \tag{6}$$

where I_K is the K-dimensional identity matrix. This objective encourages the spatial directions to be orthogonal and unit-normalized, as illustrated in Fig. 2(a).

Given a mini-batch of N latent transitions, we first center the transitions within the mini-batch:

$$\tilde{\Delta z}_i = \Delta z_i - \frac{1}{N} \sum_{j=0}^{N-1} \Delta z_j, \qquad i = 0, \ldots, N-1. \tag{7}$$

Centering removes the mean transition within the mini-batch and allows the learned directions to capture variation across latent transitions. Because the same mean is removed from every transition, centering does not change their relative differences. We then project each centered transition onto the learned spatial directions:

$$\Delta z_i^{\text{proj}} = V^\top \tilde{\Delta z}_i, \qquad \Delta z_{i,k}^{\text{proj}} = v_k^\top \tilde{\Delta z}_i, \qquad k = 0, \ldots, K-1. \tag{8}$$

Here, $\Delta z_i^{\text{proj}} \in \mathbb{R}^K$ is the projected latent transition, while $\Delta z_{i,k}^{\text{proj}}$ is its projection along the k-th spatial direction. This projection is illustrated in Fig. 2(b).

Directional Ranking. With orthogonality separating the learned spatial directions, we introduce directional ranking to organize projected latent transitions according to their corresponding spatial state-transition values.

For spatial direction v_k, we consider the projected latent-transition coefficients $\{\Delta z_{i,k}^{\mathrm{proj}}\}_{i=0}^{N-1}$ and their corresponding spatial state-transition values $\{\Delta s_{i,k}\}_{i=0}^{N-1}$. We use a signed variant of Rank-N-Contrast (RnC) (Zha et al., 2023) to align their directional orderings. The directional-ranking loss is

$$\mathcal{L}_{\mathrm{rank}} = \frac{1}{K}\sum_{k=0}^{K-1} \mathcal{L}_{\mathrm{sRnC}}\left(\{\Delta z_{i,k}^{\mathrm{proj}}\}_{i=0}^{N-1}, \{\Delta s_{i,k}\}_{i=0}^{N-1}\right). \tag{9}$$

In particular, for two transitions j and ℓ, the signed ranking objective encourages

$$\Delta s_{j,k} < \Delta s_{\ell,k} \quad \Longrightarrow \quad \Delta z_{j,k}^{\mathrm{proj}} < \Delta z_{\ell,k}^{\mathrm{proj}}. \tag{10}$$

Thus, the projected coefficient increases monotonically with its corresponding spatial state-transition component. Unlike unsigned distance ranking, the signed objective also fixes the orientation of each spatial direction and penalizes reversed or folded orderings. Appendix A.2 details the complete signed RnC formulation and includes an accompanying explanation.

More generally, the learned spatial directions establish a component-wise ordering correspondence (see Fig. 2(c)):

$$\Delta z_i^{\mathrm{proj}} = V^\top \tilde{\Delta} z_i \sim \Delta s_i, \tag{11}$$

where $\sim$ denotes directional ordering rather than numerical equality or proportionality.

Orthogonality and directional ranking provide complementary structure to the latent dynamics. Orthogonality makes the learned spatial directions mutually orthogonal, while directional ranking associates each direction with a spatial state-transition component and organizes projected transitions along that direction. The complete SWM training objective is

$$\mathcal{L}_{\mathrm{SWM}} = \mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{orth}}\mathcal{L}_{\mathrm{orth}} + \lambda_{\mathrm{rank}}\mathcal{L}_{\mathrm{rank}}, \tag{12}$$

where λ_{orth} and λ_{rank} control the contributions of orthogonal direction learning and directional ranking, respectively.

3.3 Spatial-Aware Latent Planning

The spatial directions learned by SWM provide coordinates for comparing latent displacements. Building on this structure, we develop *Spatial-Aware Latent Planning* (SLP), which connects local world-model predictions to compatible offline landmarks and derives a goal-directed roadmap-search heuristic from the projected landmark-to-goal displacement.

Latent Roadmap. Given an offline observation dataset $\mathcal{D}_{\mathrm{off}} = \{o_i\}_{i=0}^{M-1}$, we encode each observation to construct a set of latent landmarks:

$$\mathcal{V} = \{z_i = \mathrm{Enc}(o_i)\}_{i=0}^{M-1}. \tag{13}$$

These landmarks form the vertices of a directed roadmap $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, whose edges represent local transitions predicted by the world model.

Let $\mathcal{U}$ denote a set of candidate action blocks. Starting from landmark z_i, applying $u \in \mathcal{U}$ through the latent predictor produces

$$\hat{z}_{i,u} = \mathrm{Pred}(z_i, u). \tag{14}$$

For action blocks containing multiple actions, the predictor is applied recursively. Because $\hat{z}_{i,u}$ generally does not coincide with an offline landmark, we associate it with a roadmap vertex using the learned spatial directions.

Spatial-Aware Roadmap Connection. For the transition predicted from z_i under action block u, we define

$$\Delta\hat{z}_{i,u} = \hat{z}_{i,u} - z_i \tag{15}$$

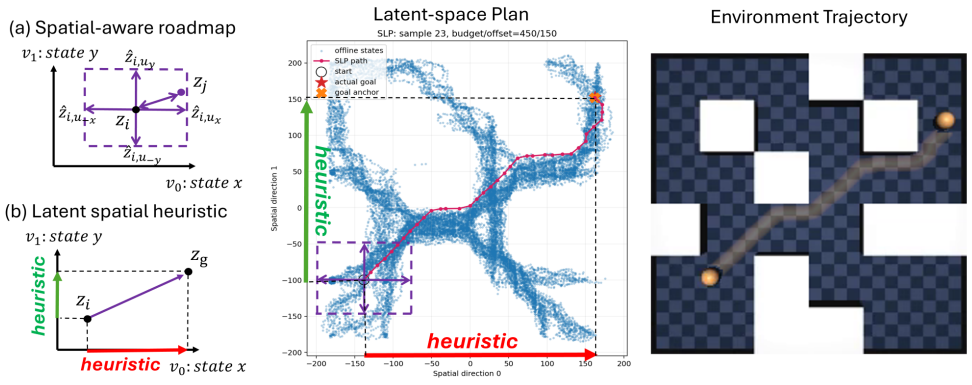


Figure 3: **Spatial-Aware Latent Planning. (a)** Projected world-model predictions are matched to compatible offline landmarks to form roadmap edges. **(b)** Projected goal displacement defines the roadmap-search heuristic. The center and right panels show the resulting latent path and executed trajectory.

and project it onto the learned spatial directions:

$$\Delta\hat{z}^{\text{proj}}_{i,u} = V^\top \Delta\hat{z}_{i,u} = V^\top(\hat{z}_{i,u} - z_i). \tag{16}$$

For each candidate landmark z_j, we similarly compute

$$\Delta z_{i,j} = z_j - z_i, \qquad \Delta z^{\text{proj}}_{i,j} = V^\top(z_j - z_i). \tag{17}$$

We retrieve the k_{nn} landmarks whose projected displacements are closest to the projected world-model prediction:

$$\mathcal{C}_{i,u} = \text{kNN}_{k_{\text{nn}}}\left(\Delta\hat{z}^{\text{proj}}_{i,u}, \{\Delta z^{\text{proj}}_{i,j}\}_{j=0}^{M-1}\right). \tag{18}$$

Among these candidates, we select the landmark closest to the prediction in the full latent space:

$$j^*(i,u) = \arg\min_{j\in\mathcal{C}_{i,u}} \|z_j - \hat{z}_{i,u}\|_2 . \tag{19}$$

The resulting connection $z_i \to z_{j^*(i,u)}$ forms a directed roadmap edge associated with action block u. The projected space identifies spatially compatible candidates, while the full latent space determines the final correspondence, as illustrated in Fig. 3(a).

Latent Spatial Heuristic. At planning time, we encode the current and goal observations as

$$z_s = \text{Enc}(o_s), \qquad z_g = \text{Enc}(o_g). \tag{20}$$

We connect z_s to the roadmap using the same transition-matching procedure and associate z_g with a nearby roadmap landmark z_{j_g}. The actual goal representation z_g is retained throughout planning rather than replaced by z_{j_g}.

For each roadmap landmark z_i, we project its displacement to the goal:

$$\Delta z^{\text{proj}}_{i,g} = V^\top(z_g - z_i). \tag{21}$$

The latent spatial heuristic is the magnitude of this projected displacement (see Fig. 3(b)):

$$h(i; z_g) = \left|\Delta z^{\text{proj}}_{i,g}\right|_2 = \left|V^\top(z_g - z_i)\right|_2 . \tag{22}$$

Because the heuristic depends only on latent representations and the learned spatial directions, it does not require the underlying spatial state during planning. Further discussion of the ordering-based nature of this heuristic is provided in Appendix A.8.

We use this heuristic to guide an A*-style search over the roadmap. Let $g(i)$ denote the accumulated roadmap cost from the current representation to landmark z_i. Its search priority is

$$\rho(i; z_g) = g(i) + \lambda_h h(i; z_g), \tag{23}$$

where λ_h controls the contribution of the heuristic. The search balances accumulated roadmap cost with projected displacement to the goal and returns a sequence of landmarks and their associated action blocks, as illustrated in Fig. 3.

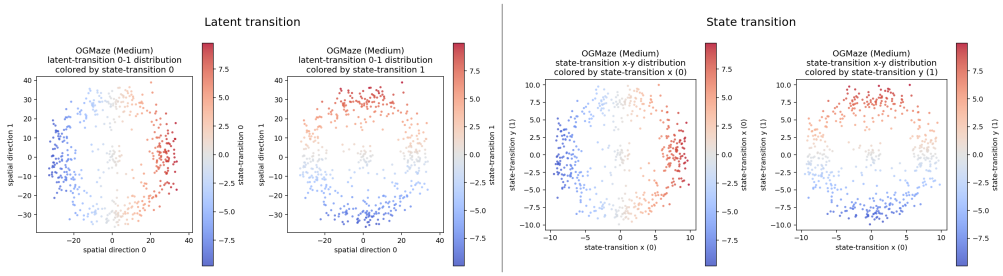


Figure 4: **Spatial directional ordering on PointMaze *Medium*.** Projected latent transitions (left) and spatial state transitions (right) are colored by the corresponding transition components, $\Delta s_0\,(\Delta x)$ and $\Delta s_1\,(\Delta y)$, respectively. Matching color progressions indicate preserved directional ordering. Additional environments are shown in Appendix A.3.

Goal Refinement. Because z_g may not coincide with an offline landmark, SLP follows the heuristic-guided roadmap until the current representation enters the goal neighborhood:

$$\left\|V^\top\left(z_g - \mathrm{Enc}(o_t)\right)\right\|_2 \le \epsilon_g. \tag{24}$$

Once this condition is satisfied, we use CEM (Rubinstein, 1997) for one execution step toward z_g, which consumes one step of the execution budget.

4 Experiments

We evaluate the *Spatial World Model* (SWM) through three hypotheses corresponding to the central claims of our method:

- H1: Does SWM learn latent dynamics that reflect the spatial organization of the environment through orthogonality and directional ordering?
- H2: Can SLP exploit SWM's learned spatial structure for effective long-horizon goal-directed planning?
- H3: Does kNN retrieval in the projected transition space accelerate latent landmark matching without degrading planning performance?

We evaluate these hypotheses on OGBench PointMaze *Medium*, *Large*, and *Giant* (Park et al., 2025), reporting latent geometry, planning performance across increasing goal offsets, and projected-space retrieval efficiency. Implementation details, extended-horizon results, and a higher-dimensional latent-structure analysis on OGBench Cube are provided in Appendices A.1, A.4, and A.6, respectively.

4.1 Evaluating State-Transition-Ranked Orthogonal Latent Dynamics

To evaluate H1, we examine the orthogonality, directional ordering, and spatial organization of the learned latent dynamics.

Orthogonality of Spatial Directions. We compute the pairwise cosine similarities between the learned spatial directions. Across PointMaze *Medium*, *Large*, and *Giant*, the off-diagonal similarities are approximately zero. This result confirms that SWM learns mutually orthogonal spatial directions. The complete cosine-similarity matrices are reported in Fig. 8 in Appendix A.3.

Directional Ordering. We evaluate whether each projected latent-transition coefficient $v_k^\top \tilde{\Delta} z$ follows the ordering of its corresponding spatial state-transition component Δs_k. Following Zha et al. (2023), we quantify this ordering using Spearman's ρ (Spearman, 1987) and Kendall's τ (Kendall, 1938). Table 2 in Appendix A.3 shows consistently strong directional ordering across all environments, with Spearman's $\rho \ge 0.992$ and Kendall's $\tau \ge 0.925$. These results confirm that the learned projections preserve the ordering of their corresponding state-transition components. Detailed coefficient–transition relationships are provided in Fig. 10 in Appendix A.3.

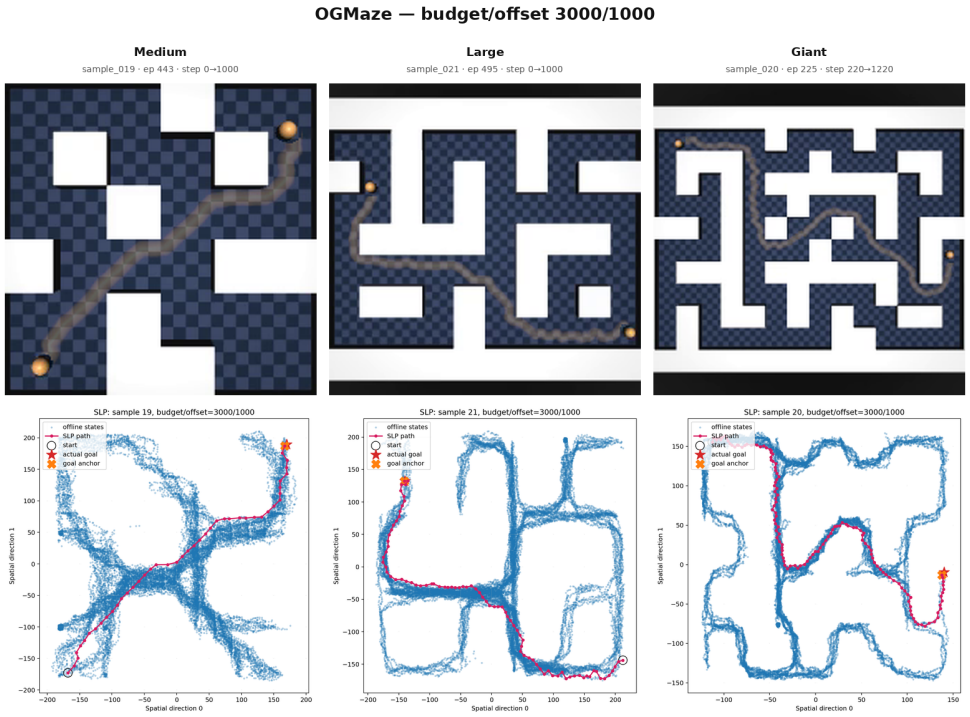


Figure 5: **Extended-horizon SLP trajectories and latent maps.** Columns show PointMaze *Medium*, *Large*, and *Giant* at an execution budget/goal offset of 3000/1000. The top row shows executed trajectories and the bottom row their corresponding paths in the learned spatial coordinates. Additional examples are provided in Appendix A.4.

Spatial Organization of Latent Dynamics. Figure 4 compares the projected latent-transition distribution with the corresponding spatial state-transition distribution. Their matching color progressions show that SWM transfers the directional ordering of spatial transitions into the learned latent dynamics. Figure 5 provides a complementary global view. The projected offline states form map-like structures that reflect environment connectivity, and the planned latent paths correspond to feasible routes through the mazes. These patterns are consistent across all three environments.

4.2 Evaluating Latent Planning and Retrieval Efficiency

Latent Planning Performance. To evaluate H2, we follow the OGBench evaluation protocol (Park et al., 2025). Figure 6 reports goal offsets of 50, 100, and 150, with the execution budget set to three times the corresponding offset. We compare SWM+SLP with DINO-WM (Zhou et al., 2024), TSLP (Wang et al., 2026b), RC-aux (Li et al., 2026), AdaJEPA (Wang et al., 2026a), and LeWM (Maes et al., 2026). We also evaluate SWM+CEM, which uses the same learned world model but replaces spatially aware roadmap planning with CEM.

Across the nine environment–offset settings, SWM+SLP achieves an average success rate of 96.1%, with individual results ranging from 90.8% to 98.4%. At offsets of 100 and 150, SWM+SLP outperforms the strongest prior method by 33.6–55.2 percentage points and, at offset 150, SWM+CEM by 25.2–33.6 points. This same-model comparison isolates the benefit of the spatial heuristic and roadmap composition for longer-horizon planning. Results extending to a goal offset of 1000 are provided in Appendix A.4.

Efficiency of Projected Landmark Retrieval. To evaluate H3, Table 1 compares landmark retrieval in the two-dimensional spatial projection with retrieval in the original 192-dimensional latent space. At an execution budget/goal offset of $450/150$, projected retrieval is $5.7\times$–$11.4\times$ faster per query and reduces total kNN time by $5.3\times$–$10.6\times$. Mean planning success also increases by 1.6–

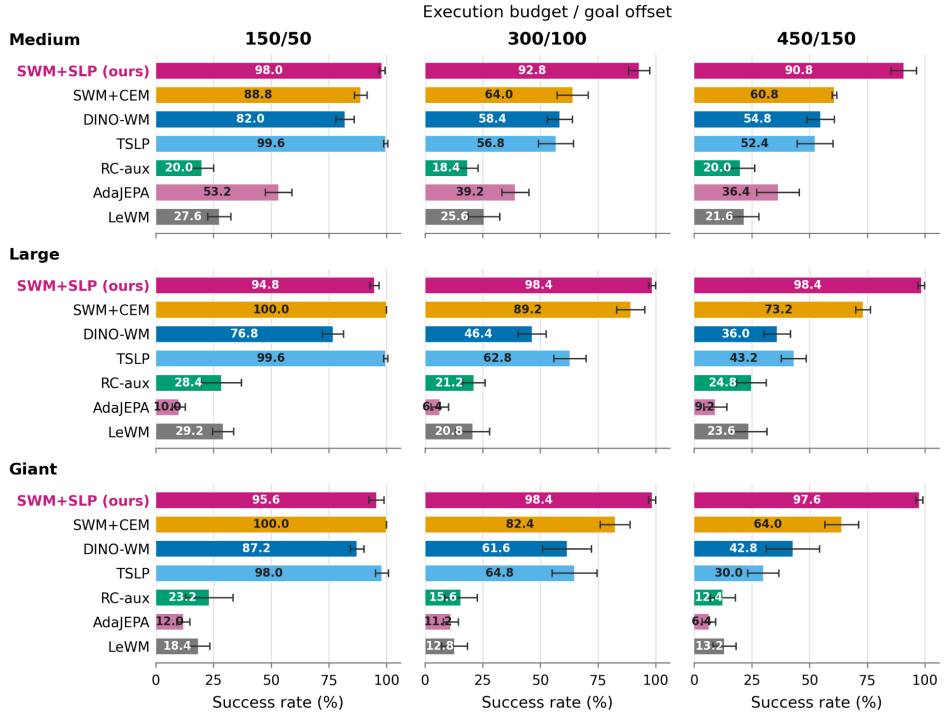


Figure 6: **Planning success on OGBench PointMaze.** Bars report mean success rates over five evaluation seeds (42–46), and error bars denote one standard deviation. Columns correspond to execution-budget/goal-offset settings, and rows correspond to maze environments.

Table 1: Projected versus full-latent kNN retrieval on OGBench PointMaze. Results are means $\pm$ standard deviations over five evaluation seeds (42–46), using 20,000 offline states at $450/150$.

Environment	Search space	Dim.	Time/query ↓	Total kNN time ↓	Success (%) ↑
Medium	Full latent	192	$178.0 \pm 43.3\,\mu s$	$23.67 \pm 5.24\,s$	89.2 ± 6.9
	Spatial projection	2	$\mathbf{15.6 \pm 0.8}\,\mu s$	$\mathbf{2.23 \pm 0.19}\,s$	$\mathbf{90.8 \pm 5.5}$
Large	Full latent	192	$112.6 \pm 3.8\,\mu s$	$15.17 \pm 0.74\,s$	94.0 ± 2.8
	Spatial projection	2	$\mathbf{15.5 \pm 0.4}\,\mu s$	$\mathbf{2.25 \pm 0.07}\,s$	$\mathbf{98.4 \pm 1.5}$
Giant	Full latent	192	$87.0 \pm 2.9\,\mu s$	$11.10 \pm 0.67\,s$	90.8 ± 3.7
	Spatial projection	2	$\mathbf{15.2 \pm 0.1}\,\mu s$	$\mathbf{2.09 \pm 0.16}\,s$	$\mathbf{97.6 \pm 1.5}$

6.8 percentage points across the three environments. These results show that the spatial projection preserves planning-relevant neighborhood structure while substantially reducing landmark-retrieval cost.

Together, these results support H1–H3: SWM learns spatially structured latent dynamics, SLP enables effective long-horizon planning, and projected-space retrieval accelerates landmark matching without degrading planning performance.

5 Conclusion

In this work, we introduced the Spatial World Model (SWM), which learns spatially structured latent dynamics through mutually orthogonal spatial directions and directional ordering of the corresponding transition coefficients. Our experiments demonstrate that the resulting latent space reflects the spatial organization of the environment while supporting effective goal-directed planning through a spatially aware latent heuristic. Overall, SWM transforms entangled and geometrically unstructured latent dynamics into spatially structured representations, connecting local world-model predictions with long-horizon planning.

AI USE STATEMENT

Generative AI tools were used to provide feedback on experimental design and presentation; check and refine mathematical notation; assist with literature retrieval, numerical aggregation, result interpretation, and figure and table preparation; and draft and revise manuscript text. All AI-assisted text, equations, citations, calculations, figures, and claims were reviewed and verified by the authors, who take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Appendix A.1 documents the model architecture, training and planning hyperparameters, dataset processing, and evaluation protocol. The remaining appendices provide complete quantitative results and additional analyses supporting the paper's claims. We plan to release the code and experiment configurations under an open-source license upon acceptance.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15619–15629, 2023.

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

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-JEPA: Latent video prediction for visual representation learning, 2024. URL https://openreview.net/forum?id=WFYbBOEOtv.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

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

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Chi-Yao Huang, Khoa Vo, Aayush Atul Verma, Duo Lu, and Yezhou Yang. Domain expansion: A latent space construction framework for multi-task learning. *arXiv preprint arXiv:2601.20069*, 2026.

Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models, 2020. URL https://arxiv.org/abs/1911.12247.

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

Martina Lippi, Petra Poklukar, Michael C. Welle, Anastasia Varava, Hang Yin, Alessandro Marino, and Danica Kragic. Enabling visual action planning for object manipulation through latent space roadmap, 2022. URL https://arxiv.org/abs/2103.02554.

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

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models, 2023. URL https://arxiv.org/abs/2209.00588.

Edvard I. Moser, Emilio Kropff, and May-Britt Moser. Place cells, grid cells, and the brain's spatial representation system. *Annual Review of Neuroscience*, 31:69–89, 2008. doi: 10.1146/annurev.neuro.31.061307.090723.

Heejeong Nam, Quentin Le Lidec, Lucas Maes, Yann LeCun, and Randall Balestriero. Causal-jepa: Learning world models through object-level latent masking, 2026. URL https://arxiv.org/abs/2602.11389.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025.

Reuven Y Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.

Tankred Saanum, Peter Dayan, and Eric Schulz. Simplifying latent dynamics with softly state-invariant world models, 2024. URL https://arxiv.org/abs/2401.17835.

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*, 588:604–609, 2020.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models, 2025. URL https://arxiv.org/abs/2502.14819.

Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 100(3/4):441–471, 1987.

Ying Wang, Oumayma Bounou, Yann LeCun, and Mengye Ren. Adajepa: An adaptive latent world model, 2026a. URL https://arxiv.org/abs/2606.32026.

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim G. J. Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning, 2026b. URL https://arxiv.org/abs/2603.12231.

Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: Learning continuous representations for regression, 2023.

Lunjun Zhang, Ge Yang, and Bradly C. Stadie. World model as a graph: Learning latent landmarks for planning, 2021. URL https://arxiv.org/abs/2011.12491.

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.

A Appendix

A.1 Implementation Details

Model Architecture. We follow the LeWM architecture (Maes et al., 2026), using ViT-Tiny as the encoder and ViT-S as the predictor. The latent representation has dimension $D = 192$. For PointMaze, we learn $K = 2$ spatial directions corresponding to x and y, with $k = 0$ and $k = 1$.

Training Hyperparameters. We train each model for 20 epochs using Adam with a learning rate of 5×10^{-5} and a batch size of 512. We set the loss weights to $\lambda_{\text{orth}} = \lambda_{\text{rank}} = 0.1$ and use a ranking temperature of $\tau_k = 1.0$ for each spatial dimension k. Model training uses random seed 42. All training and planning experiments are performed on an NVIDIA H100 GPU.

Dataset. We use the OGBench PointMaze *Medium*, *Large*, and *Giant* datasets (Park et al., 2025). We use the agent coordinates included with OGBench only to construct the state-transition labels Δs_t during training; no additional annotations are required. We form training transitions using a frame skip of 10 so that each transition captures a larger and more visually distinguishable spatial displacement. Each observation has a resolution of 224×224 pixels.

Planning Hyperparameters. We construct each latent roadmap from $M = 20{,}000$ offline landmarks. The candidate action set contains eight action blocks, each comprising 10 actions. For each predicted transition, we retrieve $k_{\text{nn}} = 10$ candidate landmarks in the projected space and select the closest candidate in the full latent space. We assign unit cost to every roadmap edge, so $g(i)$ equals the number of roadmap edges from the start node to landmark i. Because each edge represents a fixed-length action block, this cost measures the number of action blocks along the roadmap path. We set the heuristic weight to $\lambda_h = 1.0$ and switch from roadmap guidance to CEM-based goal refinement when

$$\left\| V^\top (z_g - \text{Enc}(o_t)) \right\|_2 \leq \epsilon_g,$$

where $\epsilon_g = 5.0$. Once this condition is satisfied, SWM+SLP uses CEM for one execution step toward z_g, consuming one step of the execution budget. For both the SWM+CEM baseline and this single-step refinement, we use the CEM implementation and hyperparameters of LeWM (Maes et al., 2026) without modification.

Evaluation Protocol. For each execution-budget/goal-offset setting, we evaluate 50 episodes under five sampling seeds, $\{42, 43, 44, 45, 46\}$. Each seed independently generates a set of start–goal pairs for that setting; consequently, evaluation sets differ across both seeds and goal offsets. All methods are evaluated on the same sampled episodes within each seed and setting. We report the mean and standard deviation across the five evaluation seeds.

A.2 Direction-Sign Correction for Rank-N-Contrast Loss

The original Rank-N-Contrast (RnC) loss (Zha et al., 2023) orders representation similarities according to distances between continuous targets. For spatial dimension k, its unsigned formulation compares

$$|\Delta s_{i,k} - \Delta s_{j,k}|, \qquad -\left|\Delta z^{\text{proj}}_{i,k} - \Delta z^{\text{proj}}_{j,k}\right|, \tag{25}$$

where $\Delta s_{i,k}$ is the k-th component of the state transition and $\Delta z^{\text{proj}}_{i,k} = v_k^\top \tilde{\Delta} z_i$ is its corresponding projected latent-transition coefficient.

Eq. 25 depends only on absolute differences, leaving the side of an anchor on which a coefficient lies unspecified. As a result, reversed or folded coefficient orderings can satisfy similar unsigned pairwise relations, as illustrated in Fig. 7(a,b).

We remove this ambiguity by replacing the unsigned quantities with signed differences:

$$r^{(k)}_{ij} = \Delta s_{i,k} - \Delta s_{j,k}, \qquad q^{(k)}_{ij} = -\left(\Delta z^{\text{proj}}_{i,k} - \Delta z^{\text{proj}}_{j,k}\right) = \Delta z^{\text{proj}}_{j,k} - \Delta z^{\text{proj}}_{i,k}. \tag{26}$$

For an anchor i and comparison sample j, RnC normalizes over the set

$$\mathcal{C}^{(k)}_{ij} = \left\{ \ell \in \{0, \ldots, N-1\} \setminus \{i\} \;\middle|\; r^{(k)}_{i\ell} \geq r^{(k)}_{ij} \right\}. \tag{27}$$

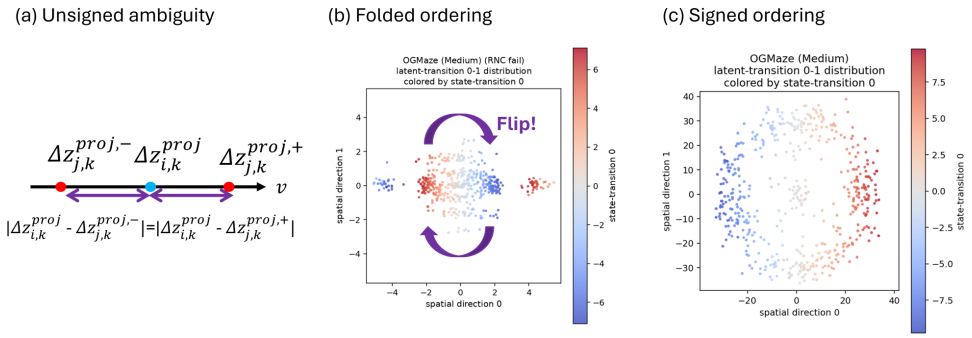


Figure 7: **Direction-sign correction for RnC. (a)** Unsigned distances do not determine which side of an anchor should contain a projected transition. **(b)** This ambiguity can produce a folded coefficient ordering. **(c)** Signed comparisons orient the spatial direction and recover a monotonic relationship between the state-transition component and its projected latent coefficient.

The corresponding pairwise loss is

$$\mathcal{L}_{ij}^{(k)} = -\log \frac{\exp\left(q_{ij}^{(k)}/\tau_k\right)}{\sum_{\ell \in \mathcal{C}_{ij}^{(k)}} \exp\left(q_{i\ell}^{(k)}/\tau_k\right)}, \tag{28}$$

where τ_k is the temperature for spatial dimension k.

The comparison condition in Eq. 27 satisfies

$$r_{i\ell}^{(k)} \geq r_{ij}^{(k)} \quad \Longleftrightarrow \quad \Delta s_{\ell,k} \leq \Delta s_{j,k}. \tag{29}$$

Minimizing Eq. 28 encourages $q_{ij}^{(k)} > q_{i\ell}^{(k)}$. Because the anchor coefficient cancels, this corresponds to

$$\Delta z_{j,k}^{\text{proj}} > \Delta z_{\ell,k}^{\text{proj}}. \tag{30}$$

Therefore, the signed objective encourages the monotonic ordering

$$\Delta s_{\ell,k} < \Delta s_{j,k} \quad \Longrightarrow \quad \Delta z_{\ell,k}^{\text{proj}} < \Delta z_{j,k}^{\text{proj}}. \tag{31}$$

This correction fixes the orientation of each learned spatial direction and penalizes reversed or folded coefficient orderings. We apply the signed RnC loss independently to the scalar coefficients of each spatial direction and average the resulting losses across directions.

A.3 Additional Analysis of Spatially Structured Latent Dynamics

We provide additional analyses of the two geometric properties underlying SWM: orthogonality and directional ordering. These results complement the quantitative rank-correlation results and latent-transition visualizations presented in the main paper.

Orthogonality of Spatial Directions. Fig. 8 reports the pairwise cosine similarities between the learned spatial directions. Across PointMaze *Medium*, *Large*, and *Giant*, the diagonal similarities equal one, while the off-diagonal similarities are approximately zero. These results confirm that SWM consistently learns mutually orthogonal spatial directions across different environment scales.

Directional Ordering. Fig. 9 compares the projected and state-transition distributions across all three PointMaze environments. Fig. 10 provides a detailed visualization of the relationship between each spatial state-transition component Δs_k and its corresponding projected latent-transition coefficient $v_k^\top \tilde{\Delta} z$. Across all environments and directions, the binned median increases consistently with the spatial state-transition value. This provides direct qualitative evidence that SWM preserves the relative ordering of spatial state transitions along their corresponding learned directions.

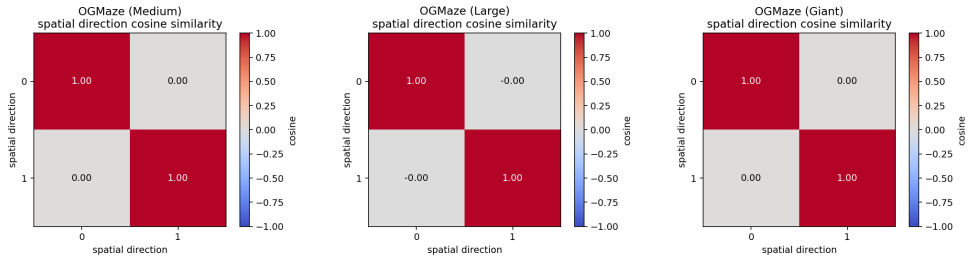


Figure 8: **Orthogonality of the learned spatial directions.** Pairwise cosine-similarity matrices of the learned spatial directions on OGBench PointMaze *Medium*, *Large*, and *Giant*. Diagonal entries equal one, while off-diagonal similarities are approximately zero, confirming that the learned spatial directions are mutually orthogonal.

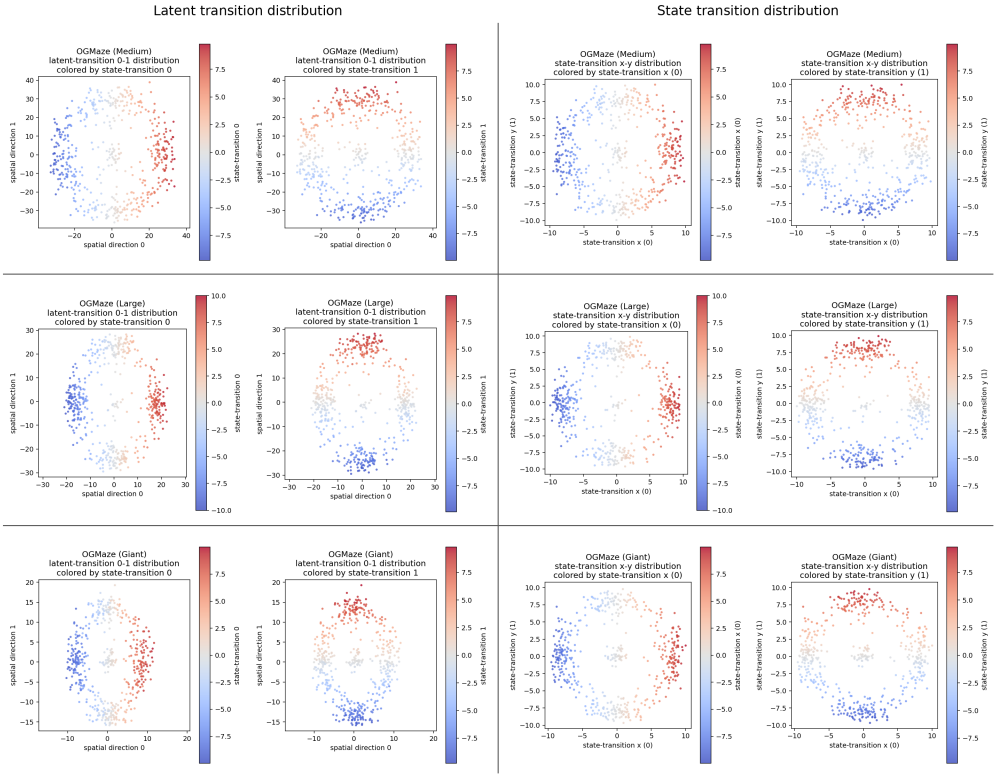


Figure 9: **Qualitative validation of spatial directional ordering.** Rows correspond to OGBench PointMaze *Medium*, *Large*, and *Giant*. The left group shows projected latent transitions $V^\top \tilde{\Delta z}$, and the right group shows the corresponding spatial state transitions $(\Delta s_0, \Delta s_1) = (\Delta x, \Delta y)$. Within each group, the two plots are colored by $\Delta s_0, (\Delta x)$ and $\Delta s_1, (\Delta y)$, respectively. The matching color progressions show that each latent direction preserves the ordering of its corresponding state-transition component.

Table 2: Directional-ordering correlations on OGBench PointMaze. Each tuple reports the correlations for the $(\Delta x, \Delta y)$ transition components, respectively. Values closer to 1 indicate stronger directional ordering.

Environment	Spearman's $\rho \uparrow$	Kendall's $\tau \uparrow$
Medium	$(0.998,\ 0.999)$	$(0.975,\ 0.974)$
Large	$(0.996,\ 0.997)$	$(0.951,\ 0.962)$
Giant	$(0.992,\ 0.996)$	$(0.925,\ 0.951)$

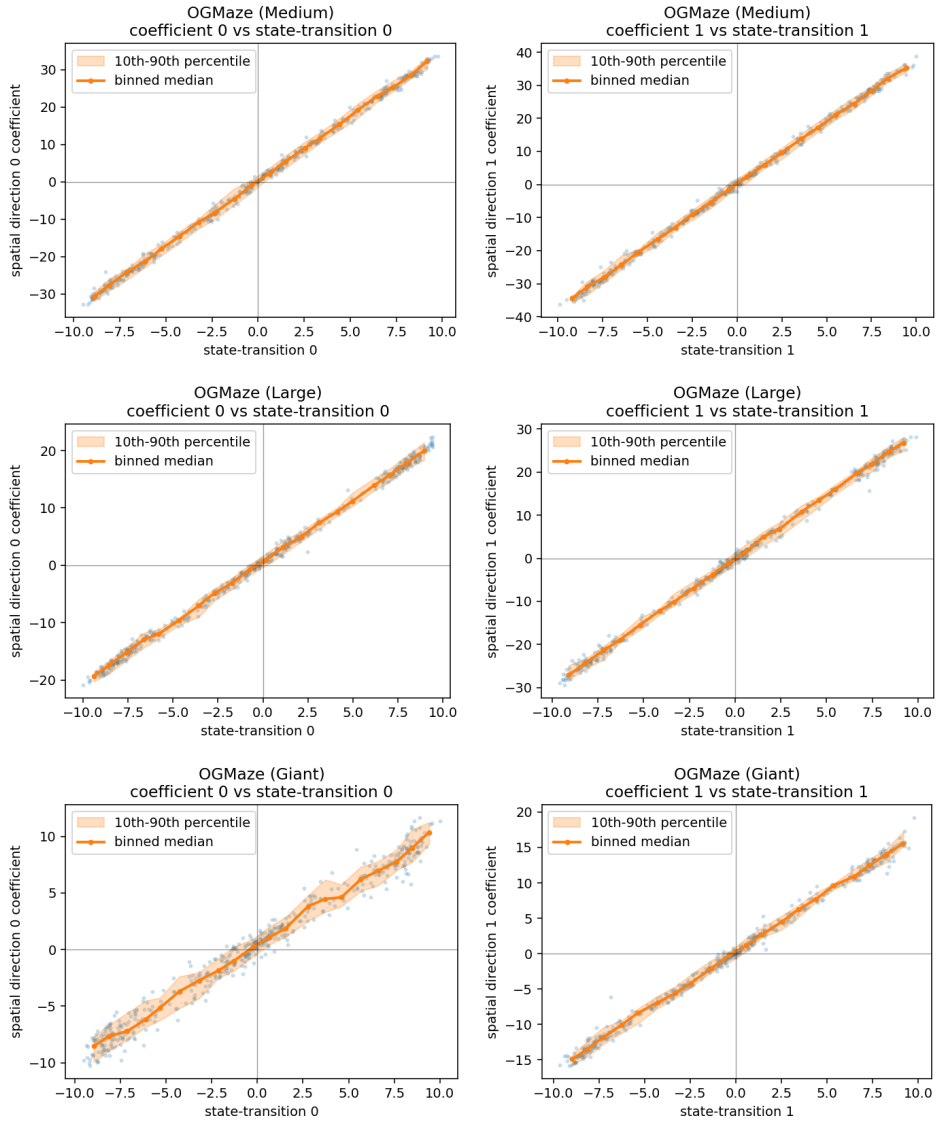


Figure 10: **Relationship between spatial state transitions and projected latent-transition coefficients.** Rows correspond to OGBench PointMaze *Medium*, *Large*, and *Giant*, while columns correspond to the x- and y-transition components, Δs_0 (Δx) and Δs_1 (Δy), respectively. Each panel plots the projected latent-transition coefficient $v_k^\top \tilde{\Delta} z$ against the associated state-transition component Δs_k. Blue points denote individual transitions, the orange curve shows the binned median, and the shaded region indicates the 10th–90th percentiles. The consistently increasing trends show that SWM preserves the ordering of each spatial state-transition component along its learned direction.

A.4 Planning over Extended Horizons

We extend the planning evaluation beyond the goal offsets considered in the main experiments using execution-budget/goal-offset settings of $750/250$, $1500/500$, and $3000/1000$. All settings maintain a budget-to-offset ratio of 3. Results are reported as mean $\pm$ standard deviation over five evaluation seeds.

Table 3 reports the complete results for the standard settings shown in Fig. 6. Table 4 shows that SWM+SLP remains effective over substantially longer horizons. At the longest setting of $3000/1000$, it achieves success rates of 88.4%, 97.2%, and 92.0% on PointMaze *Medium*, *Large*, and *Giant*, respectively. Relative to $1500/500$, performance changes by only -1.2, $+0.4$, and -3.6 percentage points across the three environments. Figures 11 and 12 show example executed trajectories and their corresponding paths in the learned spatial latent maps.

Table 3: Planning success rate (%) on OGBench PointMaze. Results are reported as mean $\pm$ standard deviation across five evaluation seeds (42–46). Column labels denote execution budget/goal offset.

	PointMaze-Medium			PointMaze-Large			PointMaze-Giant		
Method	150/50	300/100	450/150	150/50	300/100	450/150	150/50	300/100	450/150
SWM+SLP (ours)	98.0 ± 1.3	92.8 ± 4.5	90.8 ± 5.5	94.8 ± 2.0	98.4 ± 1.5	98.4 ± 1.5	95.6 ± 3.2	98.4 ± 1.5	97.6 ± 1.5
SWM+CEM	88.8 ± 2.7	64.0 ± 6.8	60.8 ± 1.0	100.0 ± 0.0	89.2 ± 6.1	73.2 ± 3.2	100.0 ± 0.0	82.4 ± 6.4	64.0 ± 7.3
DINO-WM (Zhou et al., 2024)	82.0 ± 4.0	58.4 ± 5.5	54.8 ± 5.9	76.8 ± 4.6	46.4 ± 6.1	36.0 ± 5.8	87.2 ± 3.0	61.6 ± 10.6	42.8 ± 11.5
TSLP (Wang et al., 2026b)	99.6 ± 0.9	56.8 ± 7.6	52.4 ± 7.8	99.6 ± 0.9	62.8 ± 7.0	43.2 ± 5.4	98.0 ± 2.8	64.8 ± 9.7	30.0 ± 6.8
RC-aux (Li et al., 2026)	20.0 ± 5.1	18.4 ± 4.6	20.0 ± 6.2	28.4 ± 8.8	21.2 ± 4.8	24.8 ± 6.4	23.2 ± 10.3	15.6 ± 7.1	12.4 ± 5.5
AdaJEPA (Wang et al., 2026a)	53.2 ± 5.8	39.2 ± 5.8	36.4 ± 9.2	10.0 ± 2.8	6.4 ± 3.8	9.2 ± 5.0	12.0 ± 2.8	11.2 ± 3.3	6.4 ± 3.0
LeWM (Maes et al., 2026)	27.6 ± 5.0	25.6 ± 6.8	21.6 ± 6.5	29.2 ± 4.6	20.8 ± 7.3	23.6 ± 7.9	18.4 ± 5.0	12.8 ± 5.6	13.2 ± 5.0

Table 4: Extended-horizon planning success rate (%) on OGBench PointMaze. Results are reported as mean $\pm$ standard deviation across five evaluation seeds (42–46). Column labels denote execution budget/goal offset. *Note:* AdaJEPA uses three evaluation seeds (42–44).

	PointMaze-Medium			PointMaze-Large			PointMaze-Giant		
Method	750/250	1500/500	3000/1000	750/250	1500/500	3000/1000	750/250	1500/500	3000/1000
SWM+SLP (ours)	91.6 ± 4.6	89.6 ± 2.9	88.4 ± 7.1	96.8 ± 3.7	96.8 ± 1.0	97.2 ± 1.0	94.4 ± 5.9	95.6 ± 3.9	92.0 ± 5.2
SWM+CEM	58.0 ± 5.9	68.4 ± 2.3	63.0 ± 4.4	72.8 ± 3.5	72.4 ± 6.1	69.6 ± 5.0	62.4 ± 4.6	64.4 ± 4.3	57.6 ± 7.7
DINO-WM (Zhou et al., 2024)	53.2 ± 7.3	58.0 ± 4.9	47.6 ± 7.1	38.0 ± 4.9	35.2 ± 7.2	36.4 ± 2.6	30.8 ± 8.1	29.2 ± 7.0	36.8 ± 3.0
TSLP (Wang et al., 2026b)	52.0 ± 9.3	54.4 ± 3.6	34.4 ± 7.9	37.6 ± 5.2	31.6 ± 8.2	25.6 ± 6.1	25.2 ± 8.1	18.0 ± 6.5	18.0 ± 1.4
RC-aux (Li et al., 2026)	20.0 ± 4.0	21.2 ± 5.6	14.4 ± 2.6	22.8 ± 5.0	18.8 ± 2.7	12.8 ± 3.6	13.6 ± 4.3	7.2 ± 2.3	9.2 ± 4.1
AdaJEPA (Wang et al., 2026a)	47.3 ± 4.6	40.7 ± 3.1	26.0 ± 5.3	8.7 ± 3.1	4.0 ± 3.5	7.3 ± 3.1	5.3 ± 2.3	3.3 ± 1.2	3.3 ± 3.1
LeWM (Maes et al., 2026)	24.8 ± 5.8	26.4 ± 4.3	14.8 ± 6.1	22.8 ± 5.2	21.2 ± 5.2	10.8 ± 3.6	10.8 ± 5.9	9.2 ± 6.4	10.4 ± 3.8

A.5 Efficiency of Projected kNN Retrieval

Table 5 compares landmark retrieval in the learned two-dimensional spatial projection with retrieval in the original 192-dimensional latent space across the three standard planning horizons. Both variants use the same planner and offline bank of 20,000 landmarks, differing only in the space used for kNN search. For each of five evaluation seeds (42–46), we evaluate 50 episodes. Per-query latency is averaged over all queries within each evaluation, while total kNN time is accumulated over the complete evaluation.

Across all environments and planning horizons, projected retrieval reduces per-query latency by $5.6\times$–$11.4\times$ and accumulated kNN time by $5.3\times$–$10.6\times$. It also produces equal or higher mean planning success in every setting, with improvements ranging from 1.2 to 6.8 percentage points. These results show that the learned spatial projection substantially reduces landmark-retrieval cost while preserving planning-relevant neighborhood structure.

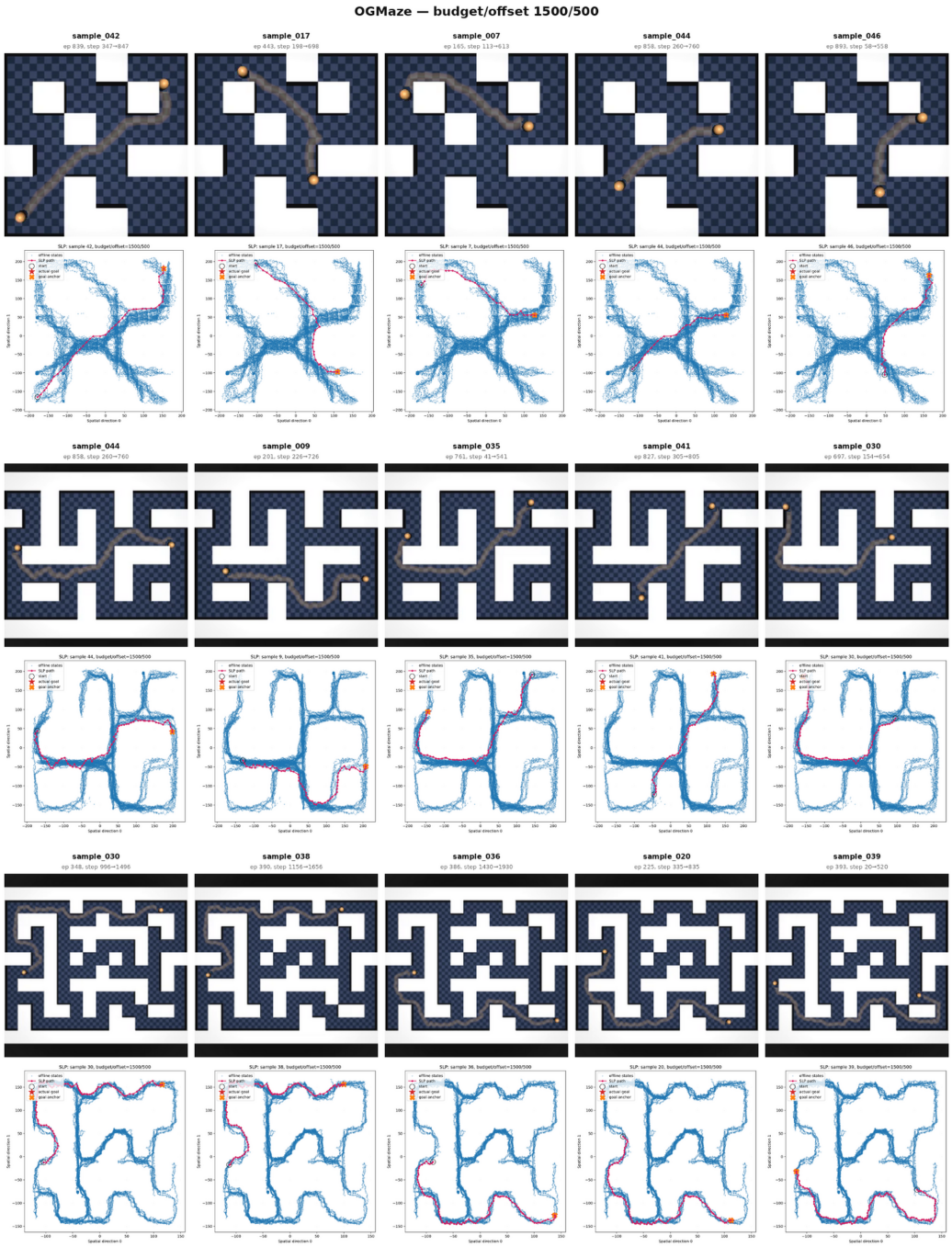


Figure 11: **Extended-horizon planning at** $1500/500$**.** Five example episodes are shown for each PointMaze environment. For each environment, the upper row shows the executed trajectories, and the lower row shows the corresponding SLP paths in the learned spatial latent map.

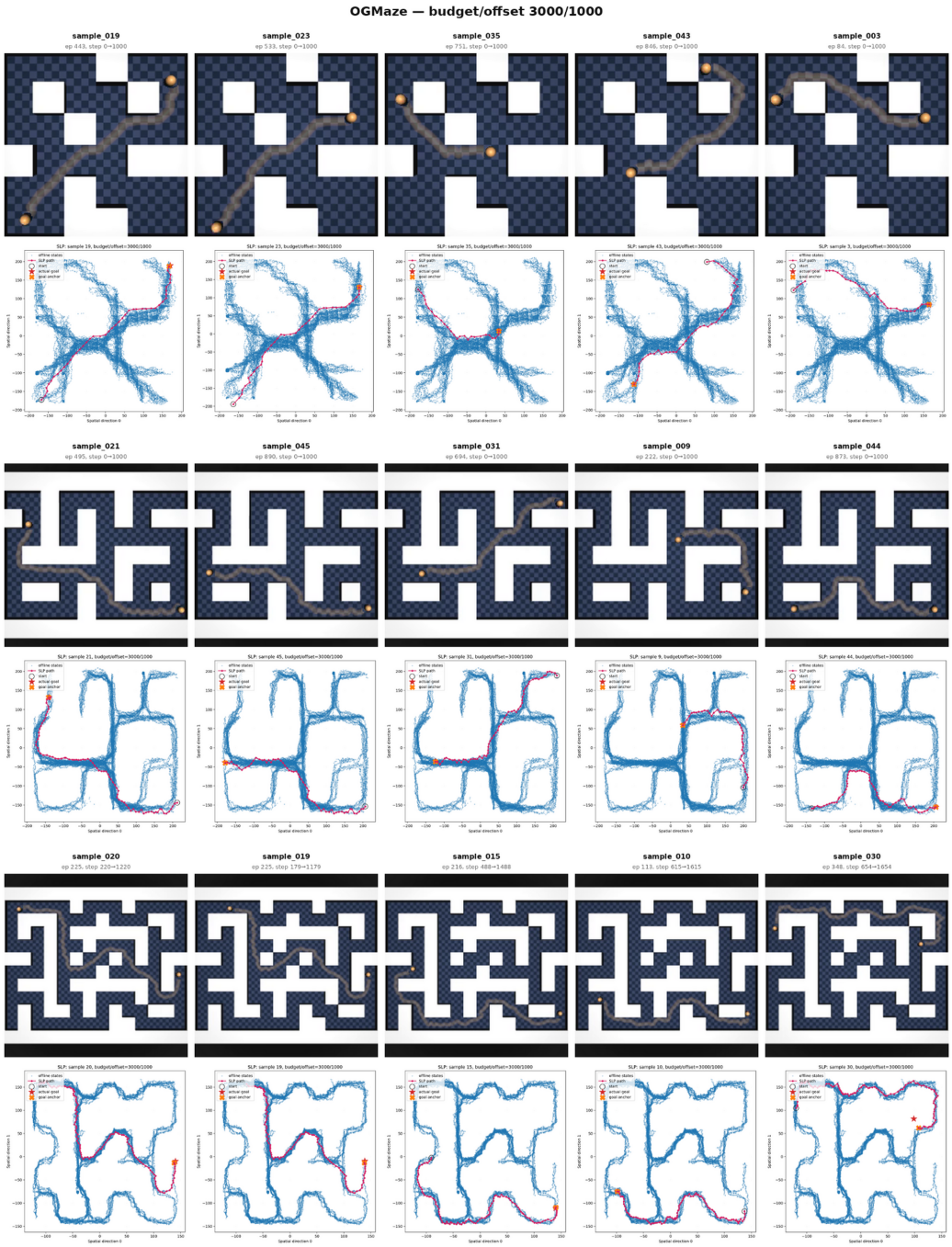


Figure 12: **Extended-horizon planning at** $3000/1000$**.** Five example episodes are shown for each PointMaze environment. For each environment, the upper row shows the executed trajectories, and the lower row shows the corresponding SLP paths in the learned spatial latent map.

Table 5: Projected versus full-latent kNN retrieval across standard PointMaze horizons. Results are mean $\pm$ standard deviation over five evaluation seeds (42–46; 50 episodes each), using 20,000 landmarks.

Environment	Budget/offset	Search space	Dim.	Time/query ↓	Total kNN time ↓	Success (%) ↑
Medium	150/50	Full latent	192	$158.8 \pm 41.6\,\mu$s	21.27 ± 5.93 s	92.4 ± 4.3
		Spatial projection	2	$\mathbf{15.4 \pm 0.1}\,\mu$s	$\mathbf{2.20 \pm 0.10}$ s	$\mathbf{98.0 \pm 1.3}$
	300/100	Full latent	192	$154.0 \pm 22.6\,\mu$s	20.65 ± 3.54 s	91.6 ± 5.3
		Spatial projection	2	$\mathbf{15.4 \pm 0.2}\,\mu$s	$\mathbf{1.96 \pm 0.42}$ s	$\mathbf{92.8 \pm 4.5}$
	450/150	Full latent	192	$178.0 \pm 43.3\,\mu$s	23.67 ± 5.24 s	89.2 ± 6.9
		Spatial projection	2	$\mathbf{15.6 \pm 0.8}\,\mu$s	$\mathbf{2.23 \pm 0.19}$ s	$\mathbf{90.8 \pm 5.5}$
Large	150/50	Full latent	192	$115.2 \pm 6.0\,\mu$s	15.51 ± 0.88 s	89.2 ± 3.7
		Spatial projection	2	$\mathbf{15.2 \pm 0.1}\,\mu$s	$\mathbf{2.24 \pm 0.05}$ s	$\mathbf{94.8 \pm 2.0}$
	300/100	Full latent	192	$112.8 \pm 3.4\,\mu$s	15.19 ± 0.69 s	92.8 ± 2.7
		Spatial projection	2	$\mathbf{15.3 \pm 0.1}\,\mu$s	$\mathbf{2.18 \pm 0.12}$ s	$\mathbf{98.4 \pm 1.5}$
	450/150	Full latent	192	$112.6 \pm 3.8\,\mu$s	15.17 ± 0.74 s	94.0 ± 2.8
		Spatial projection	2	$\mathbf{15.5 \pm 0.4}\,\mu$s	$\mathbf{2.25 \pm 0.07}$ s	$\mathbf{98.4 \pm 1.5}$
Giant	150/50	Full latent	192	$85.4 \pm 1.5\,\mu$s	10.90 ± 0.59 s	94.0 ± 4.4
		Spatial projection	2	$\mathbf{15.3 \pm 0.2}\,\mu$s	$\mathbf{1.87 \pm 0.25}$ s	$\mathbf{95.6 \pm 3.2}$
	300/100	Full latent	192	$92.9 \pm 6.6\,\mu$s	11.89 ± 1.42 s	92.4 ± 2.3
		Spatial projection	2	$\mathbf{15.3 \pm 0.3}\,\mu$s	$\mathbf{2.06 \pm 0.21}$ s	$\mathbf{98.4 \pm 1.5}$
	450/150	Full latent	192	$87.0 \pm 2.9\,\mu$s	11.10 ± 0.67 s	90.8 ± 3.7
		Spatial projection	2	$\mathbf{15.2 \pm 0.1}\,\mu$s	$\mathbf{2.09 \pm 0.16}$ s	$\mathbf{97.6 \pm 1.5}$

A.6 Higher-Dimensional Latent-Structure Analysis on OGBench Cube

PointMaze contains two-dimensional spatial transitions. To examine whether the proposed construction extends to a higher-dimensional transition space, we additionally analyze OGBench Cube using $K = 4$ learned directions corresponding to the x, y, z, and gripper transition components. This experiment focuses on the geometry of the learned representation rather than task-level performance. We do not evaluate SLP in this setting because the landmark coverage and search cost of its roadmap grow rapidly with the dimensionality of the projected space; scaling SLP through hierarchical or approximate search is left for future work.

Table 6 reports the correspondence between each task-state transition component and its projected latent-transition coefficient. The three translational components exhibit consistently high directional correlations, with Spearman's ρ ranging from 0.961 to 0.996 and Kendall's τ ranging from 0.852 to 0.954. The gripper component exhibits lower yet positive correlations of $\rho = 0.602$ and $\tau = 0.445$. Figure 13(b) shows that the underlying gripper transitions are concentrated near zero and exhibit less continuous and less linearly organized structure than the translational components, providing a weaker ordering signal. The corresponding projected distributions in Fig. 13(a) reflect this structure while preserving its relative color progression. Across the (x, y), (y, z), and $(z, \text{gripper})$ component pairs, the qualitative correspondence between the projected and task-state distributions complements the quantitative correlations. These results show that SWM can organize four task-state transition components and extend beyond the two-dimensional spatial setting considered in PointMaze.

Table 6: Directional-ordering correlations for four task-state transition components on OGBench Cube.

Transition component	Spearman's ρ	Kendall's τ
Δx	0.961	0.852
Δy	0.984	0.911
Δz	0.996	0.954
$\Delta s_{\text{gripper}}$	0.602	0.445

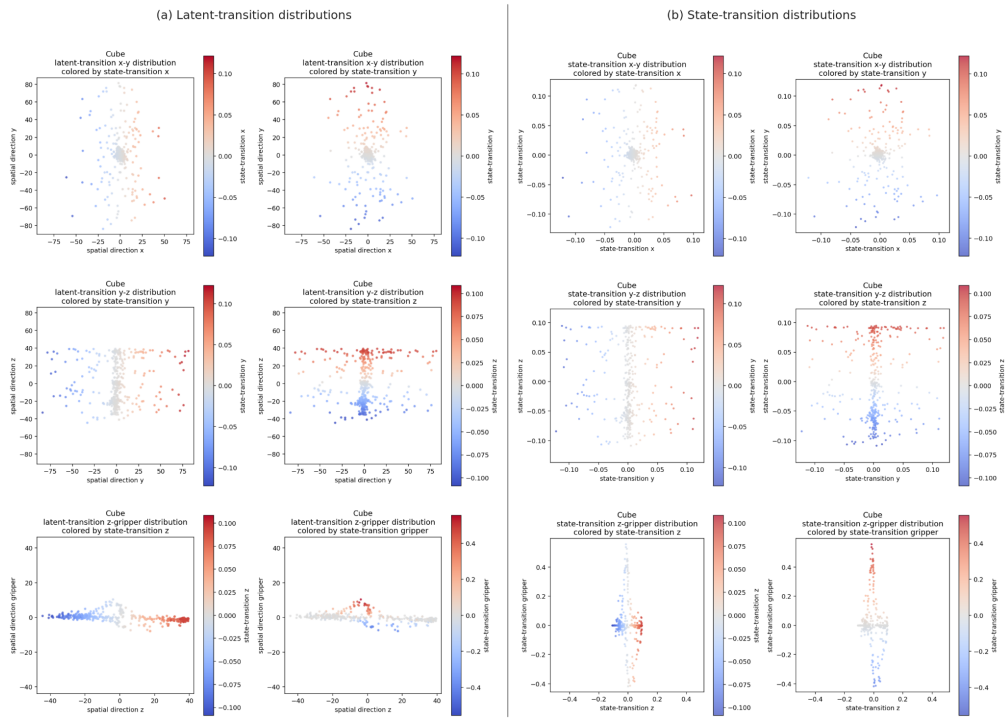


Figure 13: **Higher-dimensional directional organization on OGBench Cube. (a)** Pairwise distributions of latent transitions projected onto the learned directions. **(b)** Corresponding task-state transition distributions. Rows show the (x, y), (y, z), and $(z, \text{gripper})$ component pairs; within each row, colors denote the first and second transition components, respectively. The aligned color progressions indicate that SWM preserves directional ordering across x, y, z, and gripper transitions.

A.7 Ablation: Ranking without Orthogonal Spatial Directions

We test whether applying ranking directly to the shared latent-transition space is sufficient to learn spatially organized world-model dynamics. We train an ablated model with $\mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{rank}}$, without $\mathcal{L}_{\text{orth}}$ or separate spatial directions. Following the original RnC formulation, the ranking loss operates directly on the full-dimensional latent-transition vectors:

$$\mathcal{L}_{\text{rank}} = \frac{1}{K} \sum_{k=0}^{K-1} \mathcal{L}_{\text{RnC}} \left(\{\Delta z_i\}_{i=0}^{N-1}, \{\Delta s_{i,k}\}_{i=0}^{N-1} \right). \tag{32}$$

Thus, the ranking objectives associated with all state-transition components act on the same latent vectors Δz_i. RnC can organize latent similarities according to each continuous target without requiring projection axes; however, multiple ranking objectives must simultaneously organize the same shared representation.

We evaluate the resulting latent transitions after the same 20-epoch training schedule used for SWM. Table 7 reports rank correlations computed directly between the full latent-transition vectors and the corresponding Δx and Δy state-transition components. Both Spearman's ρ and Kendall's τ remain close to zero. Figure 14 provides the corresponding PCA visualization, in which positive and negative state transitions remain intermingled without consistent directional progressions. Although the latent transitions retain nonzero variation, the combined objective does not converge to a stable, spatially ordered latent organization.

This behavior is consistent with the collapse phenomenon discussed in Domain Expansion (Huang et al., 2026): multiple objectives acting on the same latent representation can interfere with one another and prevent the emergence of distinct factors. SWM addresses this interference by separating the state-transition components across mutually orthogonal latent directions and applying directional

ranking independently along each direction. Orthogonality and ranking therefore play complementary roles in enabling the prediction and spatial-organization objectives to converge jointly. Because the ablated world model does not converge to the intended spatial organization under the same training schedule, we do not evaluate its downstream planning performance.

Table 7: **Full-latent ranking without orthogonal spatial directions.** Epoch-20 validation rank correlations on OGBench PointMaze-Medium. Each row reports the correlation between the full latent-transition vectors Δz and the corresponding state-transition component, Δx or Δy.

State transition	Spearman's ρ	Kendall's τ
Δx	0.008	0.005
Δy	0.034	0.024

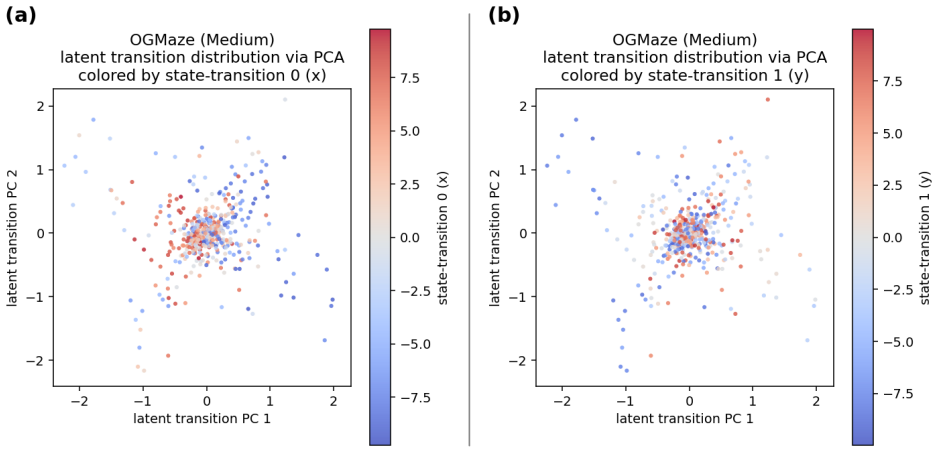


Figure 14: **Full-latent ranking without orthogonal spatial directions.** RnC is applied directly to the full latent-transition vectors Δz_i. PCA is subsequently applied to these vectors only for visualization and does not use SWM's learned spatial projection. Points are colored by **(a)** x and **(b)** y state transitions. The intermingled colors and absence of consistent directional progressions agree with the near-zero validation correlations in Table 7.

A.8 Discussion and Limitations

Ordinal Latent Heuristic. Directional ranking is designed to preserve the monotonic ordering between projected latent-transition coefficients and their corresponding spatial state-transition components, rather than to learn a numerically linear or metrically calibrated mapping. Accordingly, $h(i; z_g)$ serves as a relative reference for goal direction and proximity instead of an exact estimate of spatial distance or remaining roadmap cost. This ordered structure provides explicit directional guidance that is not readily available in conventional unstructured latent spaces, and our experiments demonstrate its effectiveness for long-horizon planning. Since the heuristic is not calibrated to roadmap cost, we use it for A*-style search without claiming admissibility, consistency, or shortest-path optimality. Learning metrically calibrated spatial directions while preserving their ordering structure is an interesting direction for future work.