How Latent Transition Geometry Matters for Planning in World Models

Anonymous authors
Paper under double-blind review

Abstract

World models support planning by predicting how the environment will evolve under candidate actions. In latent world models, the model is typically trained for one-step prediction, while planning typically requires repeatedly composing the learned transitions to compare candidate futures over a multi-step horizon. However, one-step latent predictive accuracy does not automatically translate to planning success. This raises a question: *what geometric properties of the learned one-step transition are useful for multi-step planning?* We study this question through an action-conditioned *transition-field* view of latent dynamics, using curvature regularization as a controlled geometric intervention. Across four continuous-control tasks, appropriately tuned regularization improves planning, while excessive regularization can sometimes be detrimental. To understand this observation, we decompose the transition field into an affine component and a residual non-affine component. We find that the curvature intervention reduces the field's sensitivity to action primarily by contracting its non-affine component. These sensitivity reductions propagate through the rollout, blurring the distinction among candidate futures and, in turn, altering the planner's decisions. Together, these results suggest that useful latent dynamics should retain sufficient action-induced predictive differences through multi-step propagation to keep candidate futures distinguishable to the planner.

1 Introduction

World models aim to learn predictive representations of environment dynamics for planning and control (Ha & Schmidhuber, 2018; Shen et al., 2026). Joint-embedding predictive architectures (JEPAs) provide a direct approach by predicting future states in latent space without reconstructing observations (LeCun et al., 2022). Effective planning, however, requires more than accurate one-step prediction. A planner repeatedly composes the learned transition to compare candidate action sequences, so a difference induced by an action at one step may be amplified, attenuated, or redirected before reaching the planning horizon. One-step predictive accuracy alone does not describe how these differences evolve. This raises a fundamental question: *what geometric properties of the learned transition determine whether latent predictions are useful for planning?* We approach this question by studying how action-dependent differences are organized by the learned latent dynamics and how they affect planning through multi-step rollouts.

Recent work has addressed several related aspects of this problem. Theoretical analyses characterize when JEPA representations recover latent world variables (Klindt et al., 2026) and when controlled dynamics are identifiable (Zhang et al., 2026a), while other work connects pretraining error to downstream planning regret (Cui et al., 2026) and empirically studies how model and planner choices affect physical planning performance (Terver et al., 2025). These results clarify what predictive representations can recover and how prediction quality relates to control. Here we study a complementary level of the problem: the internal geometry of the learned transition itself, and how that geometry shapes the differences among candidate futures available to a planner.

We study this question in LeWorldModel (LeWM) (Maes et al., 2026), which jointly learns a latent representation and an action-conditioned predictor $\mathcal{P}$. As illustrated in Figure 1, for a latent state z_t and action a_t, we write the predicted displacement as

$$F(z_t, a_t) = \mathcal{P}(z_t, a_t) - z_t \tag{1}$$

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.

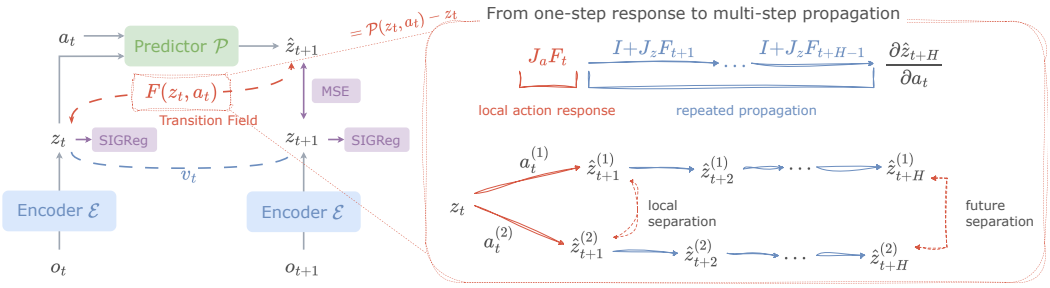


Figure 1: Overview of the transition-field view. LeWM encodes observations into latent states and predicts the next latent state under an action, which defines the transition field $F(z_t, a_t)$. During planning, this field is repeatedly composed over a rollout, so local action effects introduced through $J_a F$ are propagated by subsequent state-dependent Jacobians $I + J_z F$.

which defines an action-conditioned *transition field*. In Section 3, we formulate the learned dynamics through this field and derive three complementary geometric views: its temporal evolution along latent trajectories, its local sensitivity to action, and the propagation of action effects through multi-step rollouts. Together, these provide the mathematical basis for analyzing how action-conditioned differences are organized by the learned dynamics.

A curvature regularizer (Wang et al., 2026) acts on the encoded displacements v_t that serve as targets for the transition field, providing a natural intervention for probing its geometry. We augment LeWM with this regularizer across TwoRoom (Sobal et al., 2026), PushT (Chi et al., 2025), Reacher (Tassa et al., 2018), and Cube (Park et al., 2025). Varying its strength produces task-dependent planning behavior, while appropriately tuned regularization improves planning across all four tasks. Before using the field to analyze these regimes, we verify that it faithfully reproduces both the encoded displacements and their temporal straightness in Section 4, establishing the transition field as a reliable object for the analyses that follow.

Section 5 analyzes how curvature reshapes the transition field. Relative to an affine reference, we find that the non-affine component carries nearly all of the action-dependent response and, under stronger regularization, most of its contraction. More importantly, this reshaping is not merely descriptive. Section 6 shows that selectively modifying the non-affine differences between candidate actions changes elite selection by the cross-entropy method (CEM) (Rubinstein, 1999) and closed-loop performance. Finally, Section 7 traces these changes through the full rollout. We characterize the action-to-future geometry through *control scale*, which measures how strongly different actions can separate predicted futures, and *conditioning*, which measures how evenly those action effects are distributed across controllable directions. We also measure *distance fidelity*, which asks whether distances in latent space remain consistent with task-relevant progress toward the goal. Together, these properties explain how useful straightening reorganizes rollout geometry while preserving action-dependent variation, whereas excessive regularization suppresses differences needed for planning.

Our contributions are threefold. *First*, we formulate latent prediction as a transition field, and use it to trace how action-dependent one-step differences propagate through multi-step rollouts. *Second*, we show that curvature primarily contracts the non-affine action response, and that these changes directly alter CEM selection and planning performance. *Third*, we identify *control scale*, *conditioning*, and *distance fidelity* as complementary rollout-level properties that characterize how latent transition geometry supports planning.

2 Related Work

World models learn environment dynamics for planning, control, and simulation (Ha & Schmidhuber, 2018). Latent approaches such as PlaNet (Hafner et al., 2019b) and the Dreamer family (Hafner et al., 2019a) support planning or policy learning through imagined rollouts, while subsequent work explores discrete autoregressive dynamics with IRIS (Micheli et al., 2022), diffusion-based simulation with DIAMOND (Alonso et al., 2024), and scalable imagination training with Dream-

erV4 (Hafner et al., 2025). Recent generative systems further extend world modeling toward interactive environments and large-scale physical simulation, including Oasis (Decart et al., 2024), Genie (Bruce et al., 2024), HunyuanWorld (Team et al., 2025), and Cosmos (Agarwal et al., 2025).

Joint-embedding predictive architectures (JEPAs) learn representations by predicting target embeddings from context without requiring observation reconstruction (LeCun et al., 2022). I-JEPA (Assran et al., 2023) and V-JEPA (Bardes et al., 2023) apply this principle to masked image and video representations. Action-conditioned world models extend representation prediction to the consequences of actions: DINO-WM (Zhou et al., 2024) predicts future pretrained visual features, V-JEPA 2-AC (Assran et al., 2025) learns action-conditioned dynamics through post-training, and PLDM (Sobal et al., 2026) jointly learns representations and dynamics from reward-free offline trajectories. LeWM (Maes et al., 2026) adopts Sketched Isotropic Gaussian Regularization (SIGReg) from LeJEPA (Balestriero & LeCun, 2025) to prevent representation collapse, enabling stable end-to-end training of the encoder and predictor. Recent work further studies how model architecture, training objectives, and planner choices affect physical planning performance (Terver et al., 2025), and connects JEPA pretraining error to downstream planning regret (Cui et al., 2026).

Latent transition structure and geometry characterize how learned representations organize dynamics for prediction and control. E2C (Watter et al., 2015) and PCC (Levine et al., 2019) learn representations suited to locally linear control, while deep Koopman models (Lusch et al., 2018) seek coordinates with linear evolution. More recent theory examines when predictive representations recover underlying world structure: conditions for recovering latent world variables are characterized in LeJEPA (Klindt et al., 2026), while controlled transition identifiability is linked to conditional action variation (Zhang et al., 2026a). Complementary work encourages action-sensitive geometry by decoding actions from latent displacements so that different actions induce distinguishable latent changes (Zhang et al., 2026b). At the trajectory level, temporal straightening (Wang et al., 2026) regularizes directional changes between consecutive latent displacements, encouraging straighter representations for planning.

3 A Transition-Field Formulation

As shown in Figure 1, LeWM jointly trains an encoder $\mathcal{E}$ and an action-conditioned predictor $\mathcal{P}$, using SIGReg to prevent representation collapse. Given an observation o_t, the model computes the current latent state z_t and predicts the next latent state $\hat{z}_{t+1}$, and is trained with

$$z_t = \mathcal{E}(o_t), \qquad \hat{z}_{t+1} = \mathcal{P}(z_t, a_t), \qquad \mathcal{L}_{\text{LeWM}} = \mathbb{E}\,\|\hat{z}_{t+1} - z_{t+1}\|^2 + \lambda_{\text{sig}}\mathcal{L}_{\text{sig}}. \tag{2}$$

Latent prediction as a transition field. We rewrite next-state prediction in terms of the latent displacement $v_t = z_{t+1} - z_t$. Define

$$F(z_t, a_t) = \hat{z}_{t+1} - z_t = \mathcal{P}(z_t, a_t) - z_t. \tag{3}$$

The LeWM objective can then be written as

$$\mathcal{L}_{\text{LeWM}} = \underbrace{\mathbb{E}\,\|F(z_t, a_t) - v_t\|^2}_{\mathcal{L}_{\text{pred}}} + \lambda_{\text{sig}}\mathcal{L}_{\text{sig}}. \tag{4}$$

Thus, next-state prediction is equivalently the learning of an action-conditioned latent displacement field, which we call the *transition field*. For a fixed representation, the population optimum under squared prediction loss is $\mathbb{E}[v_t \mid z_t, a_t]$.

The dependence of this field on action determines how different actions change the predicted transition. Locally, this response is described by

$$J_a F_t = \frac{\partial F(z_t, a_t)}{\partial a_t}. \tag{5}$$

For control, however, choosing an action requires anticipating its future consequences rather than only its immediate transition. A world model supports this through planning, where candidate action sequences are rolled forward through the learned dynamics and compared according to their predicted futures. Starting from $\hat{z}_t = z_t$, the rollout follows

$$\hat{z}_{j+1} = \hat{z}_j + F(\hat{z}_j, a_j). \tag{6}$$

Holding later actions fixed, the influence of a_t on a future predicted state is

$$\frac{\partial \hat{z}_{t+H}}{\partial a_t} = \big(I + J_z F_{t+H-1}\big) \cdots \big(I + J_z F_{t+1}\big) J_a F_t, \tag{7}$$

where $F_j = F(\hat{z}_j, a_j)$. The action Jacobian therefore determines the immediate action effect, while the subsequent state Jacobians determine how that effect is propagated over the planning horizon.

Temporal straightness of the transition field. A complementary view is to examine how the field itself changes as the system moves through consecutive states and actions. Writing $F_t = F(z_t, a_t)$ and $\Delta a_t = a_{t+1} - a_t$, a first-order expansion along an encoded trajectory gives

$$F_{t+1} - F_t \approx J_z F_t \, v_t + J_a F_t \, \Delta a_t. \tag{8}$$

The first term captures variation induced by movement through latent state space, while the second captures variation induced by changes in action. We summarize this temporal organization through

$$s_t^F = \cos(F_t, F_{t+1}), \qquad s_t^v = \cos(v_t, v_{t+1}). \tag{9}$$

The two quantities describe temporal straightness at different levels: s_t^v characterizes the geometry of the encoded trajectory, while s_t^F characterizes the geometry of the learned transition field that predicts it. Because the prediction objective trains F_t toward v_t, the geometry of the encoded displacements provides a natural way to influence the geometry learned by the transition field. The corresponding derivations are provided in Appendix A.

4 Curvature Regularization as a Geometric Intervention

The transition-field formulation links the temporal geometry of encoded trajectories to the dynamics used for planning. To test whether this geometry matters, we introduce curvature regularization (Wang et al., 2026) as a controlled intervention that leaves the model architecture and planner unchanged and penalizes directional changes between consecutive encoded displacements,

$$\mathcal{L}_{\text{curv}} = \mathbb{E}[1 - s_t^v], \qquad \mathcal{L} = \mathcal{L}_{\text{LeWM}} + \lambda \mathcal{L}_{\text{curv}}. \tag{10}$$

Varying λ lets us modify trajectory straightness and examine how these changes are reflected in the learned field and downstream planning. For each task and λ, we trained three models with three independent training seeds. For the subsequent field and rollout analyses, we randomly selected one trained checkpoint for each task and λ and kept it fixed throughout. Closed-loop evaluations used independent planner randomness.

Straightening induces task-dependent planning regimes. We vary the intervention strength λ and evaluate downstream planning. Table 1 reports representative weak, intermediate, and strong settings, with the complete sweep provided in Appendix B.

Table 1: Curvature regularization produces different planning regimes across tasks.

λ	TwoRoom	PushT	Reacher	Cube
0 (Base)	89.3±4.2	94.0±4.0	86.7±2.3	72.7±3.1
0.0005	87.3±1.2	97.3±1.2	85.3±1.2	74.7±1.2
0.1	96.0±0.0	54.7±9.5	88.7±4.2	78.7±1.2
0.5	98.0±0.0	26.0±4.0	84.0±3.5	84.7±1.2

The effect is strongly task dependent. TwoRoom and Cube benefit from relatively strong regularization, Reacher favors an intermediate strength, while PushT improves only under weak regularization and deteriorates rapidly as λ increases. The same geometric intervention therefore gives rise to qualitatively different planning regimes, motivating a closer examination of what changes inside the learned dynamics as straightening becomes beneficial or excessive.

The learned field reproduces trajectory geometry. Before using the transition field F as the object of analysis, we verify that it faithfully represents the encoded dynamics in the base models.

We evaluate $F_t = F(z_t, a_t)$ and its displacement target v_t on the same 100 held-out trajectories for each task. At the transition level, we measure

$$R^2(F, v) = 1 - \frac{\sum_{i=1}^{N} \|v_i - F_i\|^2}{\sum_{i=1}^{N} \|v_i - \bar{v}\|^2}, \tag{11}$$

where i indexes the N held-out transitions and $\bar{v}$ is their mean displacement. Using the straightness measures defined in Equation 9, we further compute

$$S_v = \mathbb{E}[s_t^v], \qquad S_F = \mathbb{E}[s_t^F], \qquad \rho_{\mathrm{str}} = \mathrm{corr}(s_t^F, s_t^v). \tag{12}$$

Here, $|S_F - S_v|$ asks whether field and displacement straightness agree on average, while ρ_{str} asks whether they vary together across individual adjacent-transition pairs.

Table 2: The learned base transition field closely reproduces encoded displacements and their temporal straightness on 100 held-out trajectories.

Metric	TwoRoom	PushT	Reacher	Cube
$R^2(F, v) \uparrow$	0.995	0.981	0.997	0.993
$\lvert S_F - S_v \rvert \downarrow$	0.005	0.030	0.004	0.012
$\rho_{\mathrm{str}} \uparrow$	0.984	0.934	0.998	0.992

As shown in Table 2, the base field closely reproduces both the encoded displacement and its temporal straightness across all four tasks. The high R^2, small $|S_F - S_v|$, and high ρ_{str} establish the transition field as a faithful representation of the learned dynamics. We therefore analyze F directly under the curvature intervention to understand what distinguishes the planning regimes above.

5 How Does the Curvature Intervention Reshape the Field?

Planning compares candidate action sequences initialized from the same current state, and Equation 7 shows that their differences first enter the rollout through $J_a F_t$ before being propagated by the subsequent state dynamics. We therefore begin by examining how the curvature intervention changes the field's local response to action.

Curvature contracts action-dependent field response. At a fixed latent state, changing the action can alter both the magnitude and the direction of the predicted transition. We summarize these effects using total and directional action sensitivity,

$$S_a = \mathbb{E}\left[\|J_a F\|\right], \qquad S_d = \mathbb{E}\left[\|J_a \hat{F}\|\right], \tag{13}$$

where $\hat{F}$ denotes the unit direction of F. Thus, S_a measures how strongly the transition vector changes with action, while S_d measures how strongly its direction changes, discounting pure changes in magnitude. To compare regularization strengths within each task, we report contraction relative to the base model,

$$C_a^\lambda = 1 - \frac{S_a^\lambda}{S_a^0}, \qquad C_d^\lambda = 1 - \frac{S_d^\lambda}{S_d^0}. \tag{14}$$

Figure 2 shows how total and directional action contraction evolve across the full regularization sweep. Across all four tasks, stronger curvature regularization generally produces greater contraction of the action-dependent field response, with small local non-monotonicities.

The non-affine component carries the contraction. The full-field measurements above show that action response contracts, but not which part of the learned dynamics carries this change. We first ask how much of the transition field can be described by a single global affine controlled-dynamics model. Let $z_t \in \mathbb{R}^d$ and $a_t \in \mathbb{R}^q$. For each task and checkpoint, we fit

$$L(z_t, a_t) = Az_t + Ba_t + b, \qquad A \in \mathbb{R}^{d \times d}, \quad B \in \mathbb{R}^{d \times q}, \quad b \in \mathbb{R}^d, \tag{15}$$

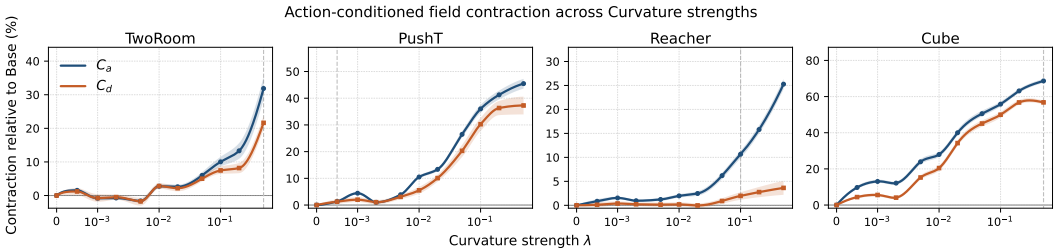


Figure 2: Curvature regularization systematically contracts both total and directional action sensitivity across all four tasks, with stronger regularization generally producing larger contraction.

using all valid transitions from 500 fitting trajectories and evaluate the fitted map on 100 held-out trajectories. Although the held-out fits vary across tasks, all leave substantial transition structure unexplained by the global affine model. We therefore define the remaining non-affine component

$$G(z_t, a_t) = F(z_t, a_t) - L(z_t, a_t), \qquad J_a F = B + J_a G. \tag{16}$$

The affine reference has a fixed action Jacobian $J_a L = B$, whereas $J_a G$ captures the action-dependent response that cannot be represented by a single global affine action map.

We measure how much of the action response is carried by this component using

$$Q_G^\lambda = \frac{\mathbb{E}\left[\|J_a G^\lambda\|^2\right]}{\mathbb{E}\left[\|J_a F^\lambda\|^2\right]}, \qquad C_G^\lambda = 1 - \frac{\mathbb{E}\left[\|J_a G^\lambda\|^2\right]}{\mathbb{E}\left[\|J_a G^0\|^2\right]}. \tag{17}$$

Here, Q_G^λ measures the fraction of action-response energy carried by the non-affine component, while C_G^λ measures its contraction relative to the base model.

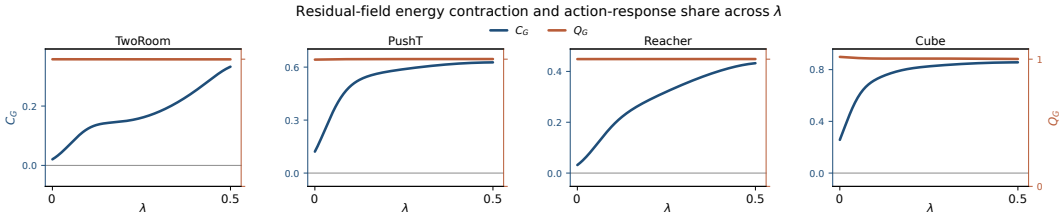


Figure 3: The non-affine component consistently carries nearly all action-response energy across the curvature sweep, while its magnitude contracts systematically as regularization increases.

As shown in Figure 3, Q_G^λ remains close to one, while C_G^λ generally increases with regularization strength. Held-out affine-fit statistics and complete attribution results are provided in Appendix C. Curvature regularization therefore contracts the field's action-dependent response primarily through its non-affine component. Since different actions produce different predicted transitions, changing this response can change how the planner compares candidate actions.

6 How Does the Reshaped Field Affect CEM?

We use CEM for planning. At each planning step, it evaluates 300 candidate action sequences, retains the top 30 elites, and updates the proposal over 25 iterations with a horizon of 5.

Modulating non-affine action response. We first test this connection at the initial elite-selection step by modulating candidate-dependent differences while preserving the native transition at the CEM proposal center. For a fixed latent state z and proposal mean $\bar{a}$,

$$F(z, a) = F(z, \bar{a}) + B(a - \bar{a}) + \big[G(z, a) - G(z, \bar{a})\big], \tag{18}$$

where $F = L + G$ and $L(z, a) = Az + Ba + b$. We define

$$F_\beta(z, a) = F(z, \bar{a}) + B(a - \bar{a}) + \beta\big[G(z, a) - G(z, \bar{a})\big]. \tag{19}$$

Thus, $\beta = 1$ recovers the native field, $\beta < 1$ contracts the non-affine differences across candidate actions, and $\beta > 1$ amplifies them. At $\beta = 0$, the proposal-centered transition and affine action difference are retained, while the candidate-dependent non-affine difference is removed. Holding z and $\bar{a}$ fixed,

$$J_a F_\beta(z, a) = B + \beta J_a G(z, a), \tag{20}$$

so β directly controls the non-affine action response.

We then test whether this intervention changes CEM selection. For each held-out state, we sample the same population of $N = 300$ candidate action sequences from the initial proposal and evaluate it under every β. Let $\mathcal{E}_\beta$ denote the top-K elite set, with $K = 30$. Relative to the native field at $\beta = 1$, we measure

$$O_\beta = \frac{|\mathcal{E}_\beta \cap \mathcal{E}_1|}{K}, \tag{21}$$

which measures how much of the native elite set is preserved under the intervention.

The intervention affects downstream planning. We next run the complete planner under $\beta \in \{0, 0.25, 0.5, 0.75, 1, 1.25, 1.5\}$ for the base and selected curvature checkpoint of each task, together with the over-regularized PushT checkpoint at $\lambda = 0.1$. Planning performance is reported relative to the native field at $\beta = 1$, in percentage points, with positive values indicating degradation and negative values indicating improvement.

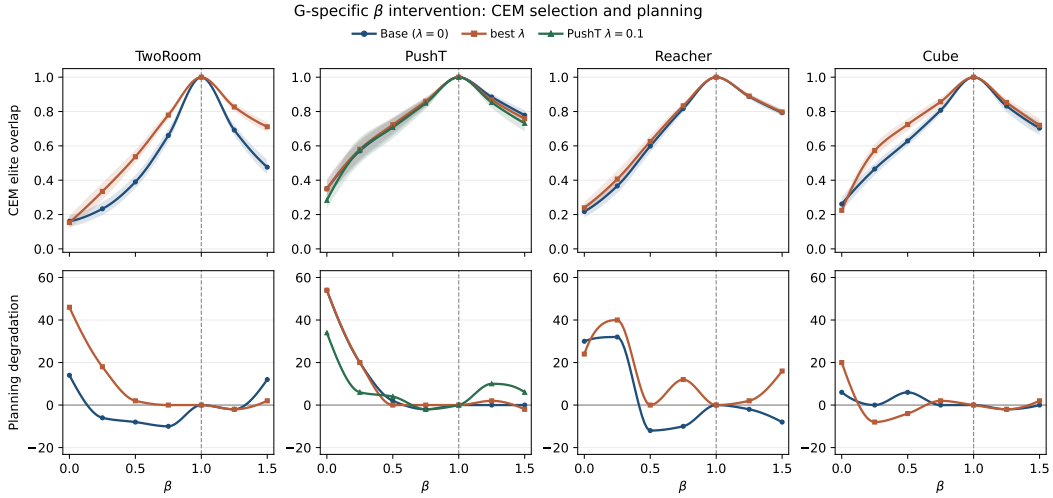


Figure 4: Modulating the non-affine action response changes both CEM elite selection and closed-loop planning. Top: elite-set overlap with the native field at $\beta = 1$. Bottom: planning-performance change relative to the native field.

Figure 4 shows that changing only the non-affine candidate-dependent response changes which action sequences CEM retains, and that these changes carry through to closed-loop planning performance. Removing this response at $\beta = 0$ substantially alters the native elite set and degrades the high-performing regimes. The dependence on β is not monotonic, however, so larger non-affine differences are not uniformly better. Rather, the intervention shows that this component carries distinctions among candidate actions that are actively used by CEM.

This field intervention links the action-dependent structure identified in Section 5 to candidate selection during planning. We further trace how these action-dependent differences are transformed through the full predicted rollout.

7 How Does Transition Geometry Enter Planning?

Planning depends not only on how actions create local differences in the transition field, but on how those differences survive the rollout and how the resulting futures are evaluated relative to the goal.

We therefore examine two complementary parts of planning geometry: the action-to-future map induced by repeated composition of F, and the latent distance used to score those futures.

From local response to action-to-future geometry. Equation 7 shows how the effect of a single action is transformed by subsequent state dynamics. Collecting the effects of the complete action sequence gives the terminal action Jacobian

$$\mathcal{J}_H = \frac{\partial \hat{z}_{t+H}}{\partial(a_t, \dots, a_{t+H-1})}. \tag{22}$$

Because CEM explores action sequences according to its proposal covariance Σ_A, we evaluate this map at the scale of the perturbations actually proposed by the planner,

$$\widetilde{\mathcal{J}}_H = \mathcal{J}_H \Sigma_A^{1/2}. \tag{23}$$

Let $\sigma_1 \geq \cdots \geq \sigma_r > 0$ denote the nonzero singular values of $\widetilde{\mathcal{J}}_H$. We summarize the resulting action-to-future geometry by

$$K_{\text{eff}} = \left(\frac{\sigma_1}{\sigma_r}\right)^2, \qquad C_H = \left(\sum_{i=1}^{r} \sigma_i^2\right)^{1/2}. \tag{24}$$

Conditioning, measured by K_{eff}, describes how unevenly proposal-scale action variation is mapped across controllable future directions; we report $\log K_{\text{eff}}$, with lower values indicating more balanced directions. *Control scale*, measured by C_H, quantifies the overall magnitude of terminal-state variation induced by those action perturbations. A smaller C_H therefore means that candidate actions produce less separated terminal futures, without implying that such contraction is inherently beneficial or harmful.

Distance fidelity characterizes the planning objective. Planning also depends on whether the latent goal distance used to score predicted futures is aligned with task-relevant progress. For state i, let

$$d_i^z = \|z_i - z_g\|^2, \qquad d_i^x = d_{\text{task}}(x_i, x_g). \tag{25}$$

We measure their pairwise ordering agreement by

$$R_{\text{obj}} = \Pr\left[(d_i^z - d_j^z)(d_i^x - d_j^x) > 0\right], \tag{26}$$

over state pairs sharing the same goal, excluding ties in task distance. Higher R_{obj} means that latent goal distance more faithfully preserves task-space progress. The full rollout derivation, linear-dynamics reduction, numerical details, and distance-fidelity evaluation are provided in Appendix D.

Geometry reveals task-dependent planning regimes. Figure 5 reveals distinct task-dependent regimes. TwoRoom and Cube show the clearest common pattern: as λ increases, conditioning improves and distance fidelity rises while control scale contracts, accompanied by steadily improving planning. PushT behaves differently. Under weak regularization, conditioning and control scale remain close to their base values while distance fidelity improves slightly, together with a small planning gain. As λ increases, control scale contracts sharply and distance fidelity returns toward its base level, while planning rapidly deteriorates. Since control scale also contracts in the other tasks, contraction itself is not inherently harmful; its useful range is task dependent. In PushT, the stronger contraction is consistent with Sections 5 and 6, where reduced action-dependent response alters the candidate distinctions used by CEM. Reacher exhibits a non-monotonic pattern. At small λ, conditioning and control scale decrease while distance fidelity worsens and planning improves. At intermediate λ, the decreases in conditioning and control scale flatten while fidelity continues to decline and planning drops. At larger λ, conditioning and control scale decrease again and planning recovers despite the continued fidelity loss.

Taken together, the transition field provides a hierarchy from local action response to planner-visible futures. The local Jacobian $J_a F$ determines how actions create transition differences, while the subsequent state dynamics propagate and reorganize these differences over the rollout. Their composition determines the scale and conditioning of the action-to-future geometry exposed to the planner, while distance fidelity characterizes whether the latent goal distance used to evaluate those futures is aligned with task-space progress. **Planning utility therefore depends jointly on how action-dependent differences are organized across predicted futures and how those futures are evaluated relative to the goal.**

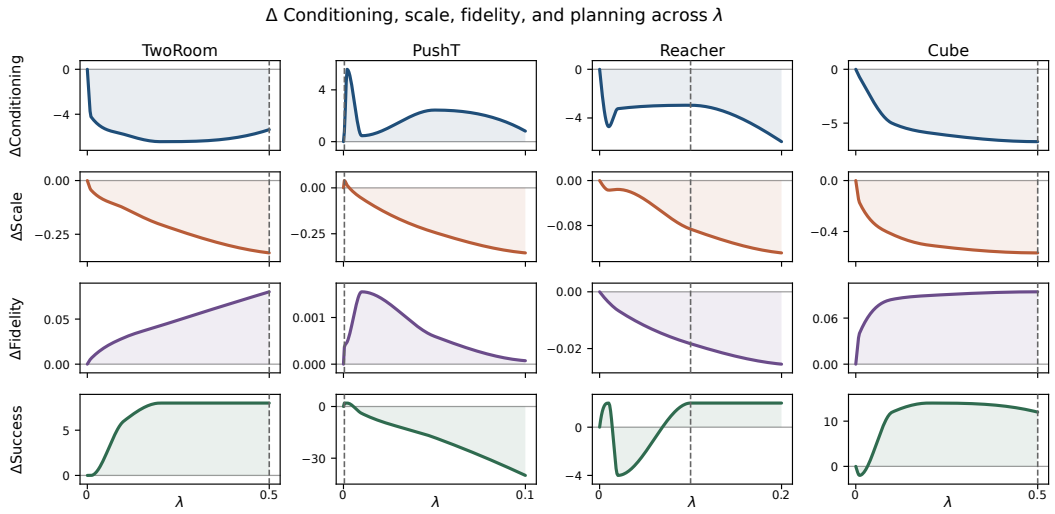


Figure 5: Transition geometry and planning across curvature strengths. Columns correspond to tasks, with all quantities relative to the base model. ΔConditioning is the change in $\log K_{\text{eff}}$ (negative means better conditioned), ΔControl Scale the relative change in C_5 (negative means contraction), ΔDistance Fidelity the change in R_{obj}, and ΔSuccess the change in planning success in percentage points. Dashed lines mark the best-performing evaluated λ for each task. Closed-loop planning is stochastic due to CEM sampling.

8 Discussion

We study how the latent geometry matters for planning in world models. Viewing the predicted displacement as a transition field links one-step prediction to the way action effects propagate through multi-step rollouts. Using curvature regularization as a controlled intervention, we find that stronger regularization systematically reduces the field's sensitivity to action, with most of this contraction occurring in its non-affine component. Intervening on these action-dependent differences changes CEM's elite selection and closed-loop performance. As the local changes are repeatedly propagated through the rollout, they reshape the differences among candidate futures seen by the planner. Overall, our results suggest that useful latent dynamics preserve action-induced differences through multi-step composition, keeping candidate futures sufficiently distinguishable for planning.

Our analysis is limited to LeWM, four controlled tasks, and local Jacobian-based characterizations around observed states and planner rollouts. Whether the same geometric structure extends to larger world models, longer horizons, or broader classes of planners remains open. Curvature regularization is also only one way to intervene on transition geometry. A natural next step is to design objectives that act more directly on planner-visible rollout geometry, with the goal of learning latent dynamics that remain useful under multi-step composition.

AI Use Statement

In this work, we used generative AI tools to assist in developing and refining conceptual frameworks and hypotheses, formulating mathematical claims and derivations, designing research methodology and experiments, implementing methods, and interpreting experimental results. We did not use generative AI tools to generate synthetic datasets, translate research materials, clean or reformat datasets, or conduct qualitative or thematic data analysis. Tasks related to formal mathematical proofs were not applicable to this work. Additionally, we used generative AI tools for code assistance, literature search and analysis, brainstorming, drafting, and editing the manuscript. All AI-assisted work was reviewed by the authors, with code tested for correctness and written content checked against the relevant sources and experimental results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Reproducibility Statement

This work is primarily an empirical analysis of learned latent dynamics. The main text and appendices define the curvature intervention and all transition-field and rollout-geometry analyses. Detailed training settings, checkpoint selection, planner configuration, and analysis protocols are provided in Appendix E.

References

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

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, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

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

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. 2023.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first international conference on machine learning*, 2024.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Jingyi Cui, Qi Zhang, Hongwei Wen, and Yisen Wang. A generalization theory for jepa-based world models. *arXiv preprint arXiv:2606.27014*, 2026.

Etched Decart, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. *URL: https://oasis-model. github. io*, 2(3):6, 2024.

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

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

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, 2019b.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025.

David Klindt, Yann LeCun, and Randall Balestriero. When does lejepa learn a world model? *arXiv preprint arXiv:2605.26379*, 2026.

Yann LeCun et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Nir Levine, Yinlam Chow, Rui Shu, Ang Li, Mohammad Ghavamzadeh, and Hung Bui. Prediction, consistency, curvature: Representation learning for locally-linear control. *arXiv preprint arXiv:1909.01506*, 2019.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

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

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

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

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999.

Qiuhong Shen, Shihua Zhang, Yue Liao, Qi Li, Zhenxiong Tan, Shizun Wang, Shuicheng Yan, and Xinchao Wang. World action models: A survey. *arXiv preprint arXiv:2606.20781*, 2026.

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

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

HunyuanWorld Team, Zhenwei Wang, Yuhao Liu, Junta Wu, Zixiao Gu, Haoyuan Wang, Xuhui Zuo, Tianyu Huang, Wenhuan Li, Sheng Zhang, et al. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. *arXiv preprint arXiv:2507.21809*, 2025.

Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *arXiv preprint arXiv:2512.24497*, 2025.

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

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015.

Xiangteng Zhang, Yang Guan, Bo Zhang, Ya-Qin Zhang, and Shengbo Eben Li. On the identifiability of controlled world models. *arXiv preprint arXiv:2607.22430*, 2026a.

Zhenghao Zhang, Yuanxiang Wang, Zhenyu Guan, Yujia Yang, Bingkang Shi, Tianyu Zong, Hongzhu Yi, Guoqing Chao, Xingchen Chen, Tiankun Yang, et al. Delta-jepa: Learning action-sensitive world models via latent difference decoding. *arXiv preprint arXiv:2606.31232*, 2026b.

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

A.1 DISPLACEMENT FORMULATION

Let $U_t = (z_t, a_t)$. From $F(U_t) = \mathcal{P}(z_t, a_t) - z_t$ and $v_t = z_{t+1} - z_t$,

$$\begin{aligned} \mathcal{P}(z_t, a_t) - z_{t+1} &= \big(\mathcal{P}(z_t, a_t) - z_t\big) - \big(z_{t+1} - z_t\big) \\ &= F(U_t) - v_t. \end{aligned} \tag{27}$$

Hence next-state prediction and displacement-field prediction have identical errors. For a fixed encoder, consider the population prediction risk

$$\mathcal{R}(f) = \mathbb{E}\|v_t - f(U_t)\|^2. \tag{28}$$

Assuming finite second moments, define

$$m(U_t) = \mathbb{E}[v_t \mid U_t], \qquad r_t = v_t - m(U_t). \tag{29}$$

Since $\mathbb{E}[r_t \mid U_t] = 0$, conditional orthogonality gives

$$\begin{aligned} \mathcal{R}(f) &= \mathbb{E}\left\|r_t + m(U_t) - f(U_t)\right\|^2 \\ &= \mathbb{E}\|r_t\|^2 + \mathbb{E}\left\|m(U_t) - f(U_t)\right\|^2. \end{aligned} \tag{30}$$

Thus, over square-integrable measurable predictors, the population minimizer is $m(U_t) = \mathbb{E}[v_t \mid U_t]$, uniquely up to almost-sure equality.

A.2 MULTI-STEP PROPAGATION OF ACTION EFFECTS

Starting from an encoded state z_t, let $\hat{z}_t = z_t$ and consider the learned rollout

$$\hat{z}_{j+1} = \hat{z}_j + F(\hat{z}_j, a_j). \tag{31}$$

The current action first affects the next predicted state through

$$\frac{\partial \hat{z}_{t+1}}{\partial a_t} = J_a F_t, \tag{32}$$

where $F_j = F(\hat{z}_j, a_j)$. Treating subsequent actions in the candidate sequence as fixed, the sensitivity is propagated at each later step by

$$\frac{\partial \hat{z}_{t+j+1}}{\partial a_t} = \big(I + J_z F_{t+j}\big)\frac{\partial \hat{z}_{t+j}}{\partial a_t}, \qquad j = 1, \dots, H-1. \tag{33}$$

Applying the recursion gives

$$\frac{\partial \hat{z}_{t+H}}{\partial a_t} = \big(I + J_z F_{t+H-1}\big) \cdots \big(I + J_z F_{t+1}\big) J_a F_t. \tag{34}$$

Thus $J_a F_t$ introduces the local effect of the current action, while the subsequent state Jacobians propagate that effect through the rollout.

A.3 LOCAL TEMPORAL VARIATION AND FIELD STRAIGHTNESS

Along an encoded trajectory, write $F_t = F(z_t, a_t)$, $v_t = z_{t+1} - z_t$, and $\Delta a_t = a_{t+1} - a_t$. Assuming F is differentiable locally, a first-order Taylor expansion gives

$$F_{t+1} - F_t = J_z F_t \, v_t + J_a F_t \, \Delta a_t + R_t, \tag{35}$$

where

$$\|R_t\| = o(\|(v_t, \Delta a_t)\|) \tag{36}$$

as $\|(v_t, \Delta a_t)\| \to 0$.

Let $x = F_t \neq 0$, $\delta = F_{t+1} - F_t$, and

$$P_x^{\perp} = I - \hat{x}\hat{x}^{\top}, \qquad \hat{x} = \frac{x}{\|x\|}. \tag{37}$$

For $\|\delta\|/\|x\| \to 0$, a second-order expansion of the adjacent cosine gives

$$1 - \cos(x, x+\delta) = \frac{\|P_x^{\perp}\delta\|^2}{2\|x\|^2} + o\left(\frac{\|\delta\|^2}{\|x\|^2}\right). \tag{38}$$

Combining Equations 35 and 38 yields, to leading order,

$$1 - s_t^F \approx \frac{\left\|P_{F_t}^{\perp}\left(J_z F_t\, v_t + J_a F_t\, \Delta a_t\right)\right\|^2}{2\|F_t\|^2}. \tag{39}$$

Thus $1 - s_t^F$ captures the component of local field variation that changes the transition direction, whereas variation parallel to F_t changes its magnitude without changing its direction to leading order.

A.4 Relation between field and displacement straightness

The prediction error on an encoded transition is

$$\epsilon_t = F_t - v_t, \qquad F_t = v_t + \epsilon_t. \tag{40}$$

For nonzero v_t, if the relative prediction error $\|\epsilon_t\|/\|v_t\|$ is small, normalization is locally smooth and therefore

$$\frac{F_t}{\|F_t\|} = \frac{v_t}{\|v_t\|} + O\left(\frac{\|\epsilon_t\|}{\|v_t\|}\right). \tag{41}$$

Applying this relation at consecutive steps gives

$$s_t^F - s_t^v = O\left(\frac{\|\epsilon_t\|}{\|v_t\|} + \frac{\|\epsilon_{t+1}\|}{\|v_{t+1}\|}\right), \tag{42}$$

where

$$s_t^F = \cos(F_t, F_{t+1}), \qquad s_t^v = \cos(v_t, v_{t+1}). \tag{43}$$

Hence, as the relative one-step prediction errors vanish, s_t^F converges to s_t^v. This formalizes the connection between the temporal geometry of the encoded displacement targets and that of the learned transition field.

B

B.1 CURVATURE REGULARIZATION SWEEP

Table 3 reports the complete planning sweep underlying the representative regimes.

Table 3: Planning success rates (%) across curvature regularization strengths. Results are the mean ± standard deviation over three independent training seeds. Green entries indicate competitive settings whose performance is comparable to the best-performing regime when accounting for seed variation.

λ	TwoRoom	PushT	Reacher	Cube
0 (Base)	89.3±4.2	94.0±4.0	86.7±2.3	72.7±3.1
0.0005	87.3±1.2	97.3±1.2	85.3±1.2	74.7±1.2
0.001	90.7±1.2	96.7±2.3	81.3±9.9	76.7±2.3
0.002	89.3±1.2	97.3±1.2	82.0±5.3	67.3±5.0
0.005	91.3±3.1	96.7±1.2	83.3±4.2	76.7±2.3
0.01	90.0±2.0	93.3±4.2	84.0±4.0	72.0±3.5
0.02	92.0±2.0	91.3±5.8	78.7±3.1	70.7±3.1
0.05	92.7±4.2	74.7±6.4	83.3±5.0	71.3±3.0
0.1	96.0±0.0	54.7±9.5	88.7±4.2	78.7±1.2
0.2	97.3±1.2	33.3±5.0	85.3±3.1	83.3±4.6
0.5	98.0±0.0	26.0±4.0	84.0±3.5	84.7±1.2

The sweep reveals distinct task-dependent regimes. TwoRoom and Cube continue to improve under relatively strong regularization, Reacher peaks at an intermediate strength, while PushT improves only under weak regularization and degrades sharply as λ increases.

B.2 GEOMETRY UNDER REPRESENTATIVE STRAIGHTENING INTERVENTIONS

To complement the Base-model correspondence reported in the main text, Table 4 compares mean displacement and field straightness at representative planning-improving curvature settings.

Table 4: Mean encoded-displacement and field straightness for the Base checkpoints and representative planning-improving curvature settings.

	TwoRoom		PushT		Reacher		Cube	
Metric	Base	Selected-λ	Base	Selected-λ	Base	Selected-λ	Base	Selected-λ
S_v	-0.363	-0.152	0.472	0.474	-0.054	-0.005	0.221	0.772
S_F	-0.359	-0.148	0.501	0.506	-0.050	-0.002	0.233	0.781

At these representative planning-improving checkpoints, mean field straightness remains close to mean encoded-displacement straightness, including when the intervention produces substantial changes in their absolute values. This provides an additional check that the field reflects the intended geometric change in the regimes used for the subsequent analysis.

C

C.1 DIRECTIONAL ACTION SENSITIVITY

We evaluate frozen checkpoints on the same fixed 100 held-out trajectories used in the main analysis, with matched samples within each task. The implementation uses frameskip $= 5$, so a_t denotes the complete action chunk supplied to the predictor. Confidence intervals use 1000 trajectory-level bootstrap resamples, with paired resampling for base–straightened comparisons.

For the normalized field $\hat{F} = F/\|F\|$, differentiation gives

$$J_a\hat{F} = \frac{1}{\|F\|}\left(I - \hat{F}\hat{F}^\top\right) J_aF = \frac{P_{\hat{F}}^{\perp} J_aF}{\|F\|}, \tag{44}$$

where $P_{\hat{F}}^{\perp} = I - \hat{F}\hat{F}^\top$. Hence

$$S_a = \mathbb{E}\left[\|J_aF\|\right], \qquad S_d = \mathbb{E}\left[\frac{\|P_{\hat{F}}^{\perp} J_aF\|}{\|F\|}\right]. \tag{45}$$

Samples with $\|F\| \le 10^{-12}$ are excluded; none occur in the reported checkpoints.

C.2 GLOBAL AFFINE REFERENCE

For each task and checkpoint, we fit the global affine reference used in the main text,

$$L(z_t, a_t) = Az_t + Ba_t + b, \qquad A \in \mathbb{R}^{d\times d}, \quad B \in \mathbb{R}^{d\times q}, \quad b \in \mathbb{R}^d. \tag{46}$$

A single coefficient triple (A, B, b) is fitted jointly over all valid transitions from 500 fitting trajectories and evaluated on the fixed 100 held-out trajectories. Its coefficient of determination is

$$R_F^2 = 1 - \frac{\sum_i \|F_i - L_i\|^2}{\sum_i \|F_i - \bar{F}\|^2}, \qquad \bar{F} = \frac{1}{N}\sum_i F_i, \tag{47}$$

where i indexes the N valid held-out transitions.

Table 5: A single global affine reference leaves substantial transition-field variation unexplained.

Model	TwoRoom	PushT	Reacher	Cube
base	0.402	0.410	0.131	0.471
Selected-λ	0.330	0.417	0.107	0.628

Although the held-out fit quality varies across tasks, Table 5 shows that substantial transition structure remains unexplained by the global affine model in every case.

We further check whether this limitation is caused by insufficient fitting data. We repeat the affine fit using increasing numbers of fitting trajectories while retaining the full predictor context. The regression target remains the same frozen transition field, and all fits are evaluated on the same 100 held-out trajectories.

As shown in Figure 6, the fits largely stabilize by 300–500 fitting trajectories, supporting the use of 500 trajectories for the main decomposition. Within the evaluated range, additional fitting data provides limited improvement in held-out fit quality.

C.3 AFFINE–NON-AFFINE ACTION-RESPONSE ATTRIBUTION

With

$$G = F - L, \qquad J_aF = B + J_aG, \tag{48}$$

the squared action-response energy admits the exact decomposition

$$\mathbb{E}\left[\|J_aF\|^2\right] = \underbrace{\|B\|^2}_{E_L} + \underbrace{\mathbb{E}\left[\|J_aG\|^2\right]}_{E_G} + \underbrace{2\,\mathbb{E}\left[\langle B, J_aG\rangle\right]}_{E_X}. \tag{49}$$

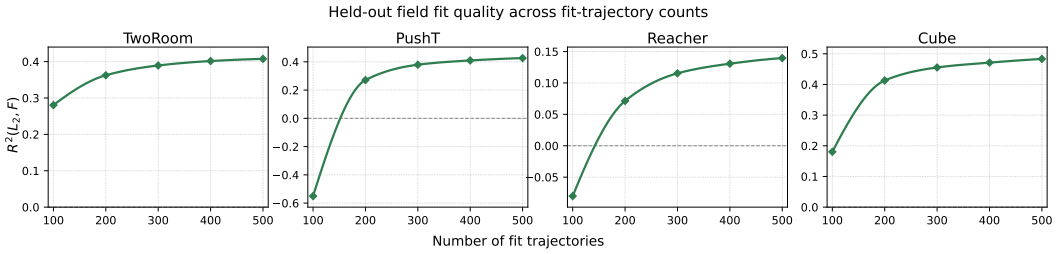


Figure 6: Held-out R^2_F of the full-context global affine reference as the number of fitting trajectories increases. Each curve corresponds to one task.

Here E_L, E_G, and E_X denote the affine, non-affine, and cross-term contributions, respectively. The energy share used in the main text is

$$Q^\lambda_G = \frac{E^\lambda_G}{E^\lambda_F}, \qquad E^\lambda_F = E^\lambda_L + E^\lambda_G + E^\lambda_X, \tag{50}$$

while

$$C^\lambda_G = 1 - \frac{E^\lambda_G}{E^0_G} \tag{51}$$

measures contraction of the non-affine action response relative to the base checkpoint. Because the cross term can be negative, Q^λ_G may slightly exceed one when the affine and non-affine action responses partially cancel.

To verify which component accounts for the change induced by curvature regularization, Table 6 attributes the base-to-selected change in full action-response energy to the three terms. Each entry reports the signed component change normalized by the corresponding change in E_F, so the rows sum to one up to rounding.

Table 6: The base-to-selected change in action-response energy is dominated by the non-affine component across all four tasks.

Component	TwoRoom	PushT	Reacher	Cube
Affine	-0.003	-0.014	0.000	0.049
Non-affine	1.001	0.977	1.001	1.030
Cross	0.003	0.038	-0.001	-0.079

The non-affine term accounts for nearly all of the base-to-selected change across tasks, while the affine and cross-term contributions remain small. Together with the Q^λ_G and C^λ_G sweep in the main text, this confirms that the curvature-induced contraction of action response is concentrated in the non-affine component.

D

D.1 MULTI-STEP ACTION-TO-FUTURE MAP

Starting from an encoded state z_t, let $\hat{z}_t = z_t$ and consider the predicted rollout

$$\hat{z}_{j+1} = \hat{z}_j + F(\hat{z}_j, a_j). \tag{52}$$

Linearizing around a reference rollout gives

$$\delta\hat{z}_{j+1} = (I + J_z F_j)\,\delta\hat{z}_j + J_a F_j\,\delta a_j, \tag{53}$$

where $F_j = F(\hat{z}_j, a_j)$. Since the current latent state is fixed during planning, $\delta\hat{z}_t = 0$. The effect of the k-th action on the terminal predicted state is therefore

$$\frac{\partial \hat{z}_{t+H}}{\partial a_{t+k}} = (I + J_z F_{t+H-1}) \cdots (I + J_z F_{t+k+1})\, J_a F_{t+k}, \qquad k = 0, \ldots, H-1, \tag{54}$$

with the empty product for $k = H-1$ equal to the identity. Collecting these blocks gives

$$\mathcal{J}_H = \left[\frac{\partial \hat{z}_{t+H}}{\partial a_t}, \frac{\partial \hat{z}_{t+H}}{\partial a_{t+1}}, \ldots, \frac{\partial \hat{z}_{t+H}}{\partial a_{t+H-1}} \right]. \tag{55}$$

Thus, $J_a F$ determines how each action creates a local transition difference, while the subsequent factors $I + J_z F$ propagate that difference to the terminal state.

For a stacked action perturbation $\delta\mathbf{a}$ with covariance Σ_A,

$$\delta\hat{z}_{t+H} \approx \mathcal{J}_H \delta\mathbf{a}, \tag{56}$$

and hence

$$\mathrm{Cov}(\delta\hat{z}_{t+H}) \approx \mathcal{J}_H \Sigma_A \mathcal{J}_H^\top. \tag{57}$$

We therefore use the proposal-whitened action-to-future map

$$\widetilde{\mathcal{J}}_H = \mathcal{J}_H \Sigma_A^{1/2}, \tag{58}$$

whose singular values describe how action perturbations at the scale explored by CEM are transformed into terminal latent variation.

D.2 CONDITIONING AND CONTROL SCALE

Let

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 \tag{59}$$

denote the numerically nonzero singular values of $\widetilde{\mathcal{J}}_H$. The two quantities used in the main text are

$$K_{\text{eff}} = \frac{\sigma_1^2}{\sigma_r^2}, \qquad C_H = \left(\sum_{i=1}^{r} \sigma_i^2 \right)^{1/2}. \tag{60}$$

K_{eff} measures spectral imbalance among the numerically controllable future directions, while C_H measures the total magnitude of proposal-scale action-induced terminal variation. Uniform rescaling changes C_H without changing K_{eff}, so scale and conditioning capture distinct properties of the action-to-future map.

For the relative quantities in Figure 5, we use

$$\begin{aligned} \Delta\mathrm{Conditioning}^\lambda &= \log K_{\text{eff}}^\lambda - \log K_{\text{eff}}^0, \\ \Delta\mathrm{ControlScale}^\lambda &= \frac{C_H^\lambda}{C_H^0} - 1. \end{aligned} \tag{61}$$

Negative $\Delta\mathrm{Conditioning}^\lambda$ indicates improved conditioning relative to the base model, whereas negative $\Delta\mathrm{ControlScale}^\lambda$ indicates contraction of the multi-step action-to-future scale.

All rollout derivatives and spectral computations are evaluated in float64. For $\widetilde{\mathcal{J}}_H \in \mathbb{R}^{m \times n}$, the numerically nonzero subspace is determined using

$$\mathrm{tol} = \max(m, n)\, \epsilon_{64}\, \sigma_1, \tag{62}$$

and σ_r is the smallest singular value above this tolerance. We use $H = 5$. For each held-out state, the same CEM proposal and sampled candidate action sequences are reused across checkpoints. Spectral quantities are computed separately for each candidate and then averaged; Jacobians are not averaged before the spectral computation.

D.3 Linear controlled dynamics as a special case

Consider the linear controlled system

$$z_{t+1} = Az_t + Ba_t. \tag{63}$$

Its transition field is

$$F(z_t, a_t) = (A - I)z_t + Ba_t, \tag{64}$$

so

$$I + J_z F = A, \qquad J_a F = B. \tag{65}$$

Substituting these quantities into Equation 55 gives

$$\mathcal{J}_H = \left[A^{H-1}B, A^{H-2}B, \ldots, AB, B\right], \tag{66}$$

which is the standard finite-horizon controllability map.

For the quadratic terminal objective

$$\ell(\mathbf{a}) = \frac{1}{2}\left\|z_{t+H}(\mathbf{a}) - z_g\right\|^2, \tag{67}$$

the rollout is affine in the stacked action sequence $\mathbf{a}$, giving

$$\nabla^2_{\mathbf{a}}\ell = \mathcal{J}_H^\top \mathcal{J}_H. \tag{68}$$

Its nonzero eigenvalues are therefore the squared singular values of $\mathcal{J}_H$. Under proposal-normalized action coordinates,

$$\delta\mathbf{a} = \Sigma_A^{1/2}\delta u, \tag{69}$$

the corresponding action-to-future map becomes

$$\widetilde{\mathcal{J}}_H = \mathcal{J}_H \Sigma_A^{1/2}. \tag{70}$$

Thus, in proposal-normalized action coordinates, the conditioning measure used in the main text reduces to the effective conditioning of the finite-horizon quadratic planning problem. The nonlinear transition-field formulation generalizes this construction by replacing powers of a fixed transition matrix with state- and action-dependent Jacobian products.

D.4 Distance fidelity

Distance fidelity characterizes the geometry used to evaluate predicted futures, but is measured directly from fixed environment state–goal pairs, independently of the learned predictor. For a pair (x_i, x_g), we encode

$$z_i = \mathcal{E}(o_i), \qquad z_g = \mathcal{E}(o_g), \tag{71}$$

and compute R_{obj} as defined in Equation 26.

For d_{task}, we use the corresponding environment-space goal error for each task, following its evaluation definition. Pairs tied under the task-space distance are excluded.

For Figure 5, we report fidelity relative to the base model,

$$\Delta R^{\lambda}_{\mathrm{obj}} = R^{\lambda}_{\mathrm{obj}} - R^{0}_{\mathrm{obj}}. \tag{72}$$

Positive values indicate improved agreement between latent goal distance and task-space progress. The same fixed state–goal pairs are used across checkpoints within each task.

E

E.1 Training Configuration

We follow the LeWM training setup. The base model and all curvature-regularized models use the same architecture and training configuration, summarized in Table 7, with the curvature regularization strength λ varied across settings.

The predictor uses a history of three observations and predicts one subsequent latent, giving a training window of four frames with a frameskip of five. All analyses use the full predictor context, including the latent and action history. For simplicity, we use z_t as the actual chunk.

Table 7: Model, training, and planning settings shared by all models.

Setting	Value
Model context	
History length	3
Prediction steps	1
Frameskip	5
Training	
Batch size	128
Optimizer	AdamW
Learning rate	5×10^{-5}
Training epochs	10
Training precision	BF16
SIGReg weight	0.09
CEM planning	
Planning horizon H	5
Candidate action sequences	300
Elites	30
Optimization iterations	25

E.2 Checkpoint Selection and Planning Evaluation

For each task and regularization strength, we randomly select one of the three training seeds and use its epoch-10 checkpoint. The selected checkpoint is kept fixed across all subsequent field analyses, rollout diagnostics, and associated planning evaluations. Closed-loop planning uses CEM with a horizon of five prediction steps, 300 candidate action sequences, 30 elites, and 25 optimization iterations. CEM sampling is randomized across evaluation runs.

E.3 Transition-Field Analysis

For each task, we use 500 trajectories for fitting and 100 held-out trajectories for evaluation. All affine fits use the full three-step predictor context, including latent and action inputs, together with an intercept. The regression targets are the corresponding outputs of the frozen transition field.

The ridge coefficient is selected by five-fold trajectory-level cross-validation within the fitting set over $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, yielding $\rho = 0.1$ for all four tasks. We then fit the affine reference on all 500 fitting trajectories and keep it fixed for held-out evaluation.

E.4 Rollout Geometry

We evaluate K_{eff} and C_H using the first 50 trajectories from the 100 held-out trajectories, with a rollout horizon of $H = 5$. At each evaluated state, we sample 16 candidate action sequences from a Gaussian reference CEM proposal with mean $\mu = 0$ and standard deviation $\sigma = 1$. The same candidate sequences are shared across all checkpoints within each task.

For each candidate, we compute the Jacobian of the terminal predicted latent with respect to the candidate action sequence through the full history-conditioned rollout. Rollout computations and forward-mode automatic differentiation are performed in float64. Spectral quantities are computed separately for each candidate before averaging. These 16 candidates are used for rollout-geometry diagnostics. Closed-loop planning uses the 300-candidate CEM configuration described above.

E.5 DISTANCE FIDELITY

We evaluate distance fidelity R_{obj} using 500 fitting trajectories. For each task, the same state–goal pairs are used across checkpoints. Latent goal distances are compared with the task-space goal distances using the pairwise ordering agreement defined in the main text.