MANIJEPA: MANIFOLD-AWARE JEPA WORLD MODELS VIA DIRECTIONAL CONTRACTION

Anonymous authors
Paper under double-blind review

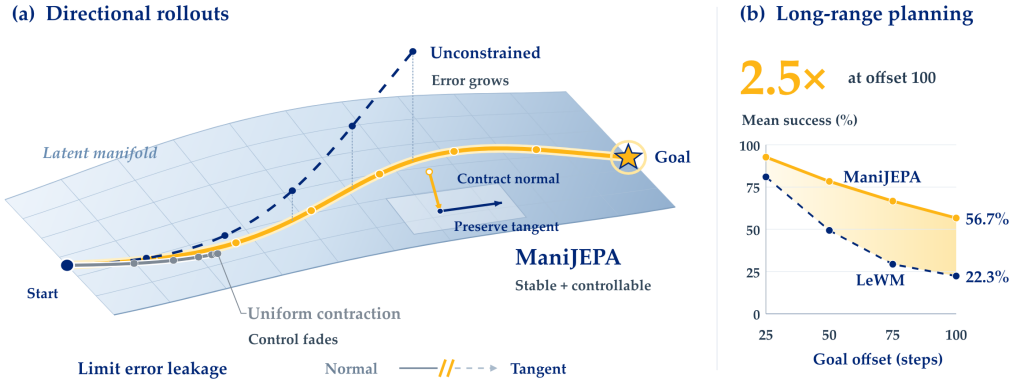

Figure 1: **Directional contraction stabilizes rollouts while preserving control authority.** **(a)** Schematic autoregressive rollouts toward a goal ($\star$). An unconstrained predictor accumulates error off the latent manifold, while uniform contraction suppresses control authority and stalls progress. ManiJEPA contracts normal directions, preserves tangent dynamics, and limits error leakage from normal to tangent directions, enabling stable rollouts toward the goal. **(b)** Planning success averaged over PushT, TwoRoom, and OGBench-Cube at increasing goal offsets. At offset 100, ManiJEPA achieves 56.7% success versus 22.3% for LeWM, approximately $2.5\times$ as high.

ABSTRACT

Joint-embedding predictive architectures learn world models by predicting future representations from current ones, enabling planning directly from pixels. But when the predictor is rolled out autoregressively at planning time, small errors compound exponentially, making long-horizon control unreliable. Existing fixes either leave the exponential form unchanged or force the predictor's Jacobian to uniformly contract, bounding error at the cost of destroying the sensitivity to early actions that planning requires. We show that this tension has a principled resolution: constrain only the directions that the training data cannot identify. Because encoder latents from reachable observations lie on a low-dimensional manifold, the predictor's behavior normal to the manifold is never observed by the teacher-forced loss and can be regularized at no bias cost, while the tangent directions that carry dynamics and control signal are preserved. ManiJEPA estimates the local tangent space of the latent manifold via local PCA on neighboring latents, decomposes the predictor Jacobian into tangent and normal blocks, and imposes three hinge penalties: normal contraction, a tangent shell that preserves on-manifold gain, and a leak term that blocks error from flowing back into the tangent space. An explicit feasibility condition coupling these penalties guarantees at most linear rather than exponential error growth, and a control-authority lower bound ensures early actions retain influence over the plan. On PushT, TwoRoom, OGBench-Cube, and LIBERO-Goal, ManiJEPA matches baselines at short horizons and substantially outperforms them at long horizons, with ablations confirming that each directional component is necessary.

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.

1 INTRODUCTION

Joint-Embedding Predictive Architectures (JEPAs) have emerged as a powerful paradigm for learning world models from high-dimensional observations (LeCun, 2022; Ha & Schmidhuber, 2018b). By predicting in representation space rather than pixel space, they avoid the intractable task of modeling irrelevant detail while capturing the structure needed for planning and control (Maes et al., 2026; Sobal et al., 2025). Recent work has demonstrated that JEPA world models can learn useful representations for robotic manipulation tasks, achieving competitive performance on short-horizon benchmarks (Maes et al., 2026; Terver et al., 2026). However, a deeper question remains: *When can we constrain a world model's predictor to prevent compounding errors without destroying the control authority needed for long-horizon planning?*

The challenge arises from a fundamental mismatch between training and evaluation. A JEPA predictor is trained on single-step transitions using teacher forcing, receiving ground-truth previous states as input. At planning time, however, the predictor is rolled out autoregressively, consuming its own predictions as inputs. This exposure bias causes error to compound: each step injects new prediction error and amplifies existing error through the predictor's Jacobian. Beyond the horizon where accumulated error exceeds goal separation, the planner optimizes noise rather than signal, making long-horizon planning impossible.

Two model-side fixes have emerged to address this compounding problem. The first, self-rollout training (Li et al., 2026b), trains the predictor on its own iterates, reducing prediction error at off-distribution inputs. While effective at shrinking the error injection term, self-rollout does not change the exponential form of error growth. The second approach, uniform Jacobian contraction (McCabe et al., 2023; Sonigra & Kumar, 2026; Li et al., 2026a), constrains the spectral norm of the predictor's Jacobian to be strictly less than one. This bounds the accumulated error but at two fundamental costs. First, control authority collapses: the sensitivity of terminal cost to early actions decays exponentially with horizon, making the planner effectively myopic. Second, on systems with conserved quantities, the true dynamics preserves certain directions, but forcing uniform contraction forgets them, imposing bias proportional to the latent scale.

Our answer is that constraints must be *directional*: targeting only the Jacobian blocks that are unidentified by the teacher-forced loss. Encoder latents from reachable observations lie on a low-dimensional reachable-state manifold. The true dynamics maps this manifold to itself, so the predictor's behavior in directions normal to the manifold is never observed by the teacher-forced loss. Constraining the normal directions is therefore regularization of an unidentified degree of freedom; it does not constrain the first-order dynamics along observed manifold directions under the stated manifold/invariance assumptions. The tangent directions, in contrast, carry the observable dynamics and must be preserved for state memory and control signal.

We close this gap with ManiJEPA, which constrains the predictor Jacobian directionally under a single explicit feasibility condition. We prove that the resulting rollout error is non-expansive while a control-authority lower bound preserves sensitivity to early actions (Propositions 1 and 2). To summarize, our contributions are:

1. A directional rollout-error bound (Proposition 1) proving that contracting only the normal Jacobian block makes the error dynamics non-expansive together with an explicit feasibility condition that is necessary and sufficient for this guarantee and whose violation makes the bound expansive.
2. A control-authority lower bound (Proposition 2) proving that tangent preservation maintains sensitivity to early actions, resolving the error-bounding versus control-authority tension.
3. An identifiability argument showing that the normal block is free to constrain (unidentified by the teacher-forced loss) while the tangent block is not, which is what makes the directional split principled rather than a heuristic regularizer.
4. An empirical validation on PushT, TwoRoom, OGBench-Cube, and LIBERO-Goal showing parity at short horizons, substantial gains at long horizons, and confirmation of the theory through rollout-error and control-authority measurements.

2 RELATED WORK

World Models. World models learn environment dynamics to let agents reason about future states in imagination (Ha & Schmidhuber, 2018b;a). Generative approaches model dynamics in pixel

space: the Dreamer lineage (Hafner et al., 2020; 2024) learns actors and critics in latent space, while IRIS (Micheli et al., 2023), DIAMOND (Alonso et al., 2024), and Genie (Bruce et al., 2024) generate high-fidelity future frames or interactive simulators. These methods typically assume reward access, jointly modeling dynamics and value. We instead target the reward-free JEPA setting, learning task-agnostic world models from observation alone.

JEPA World Models. JEPAs (LeCun, 2022) predict future states in a compact latent space rather than pixel space; I-JEPA (Assran et al., 2023) and V-JEPA (Bardes et al., 2024; Assran et al., 2025) demonstrate strong image and video representation learning through masked latent prediction. For control, early methods relied on frozen pretrained encoders (Zhou et al., 2025), capping expressivity at the pretrained features. LeWM (Maes et al., 2026) introduced end-to-end training with SIGReg regularization (Balestriero et al., 2025; Balestriero & LeCun, 2025), achieving competitive manipulation performance with a two-term loss; PLDM (Sobal et al., 2025) instead uses VICReg (Bardes et al., 2021) with temporal-straightening and inverse-dynamics heads. Hierarchical extensions (Zhang et al., 2026; Hansen et al., 2025) add multi-resolution planning but retain Euclidean latent costs. All share one limitation: the predictor is trained teacher-forced but evaluated autoregressively, so error compounds geometrically with horizon. ManiJEPA constrains the predictor Jacobian directionally to bound this error without sacrificing control authority.

Stability and Long-Horizon Planning. Error compounding in autoregressive predictors has been attacked from several sides. Self-rollout training (Li et al., 2026b) lowers injection error at off-distribution inputs but leaves the exponential growth form intact. Uniform Jacobian contraction (McCabe et al., 2023; Sonigra & Kumar, 2026) bounds accumulated error by forcing the predictor's spectral norm below one, but collapses control authority and, on systems with conserved quantities, imposes bias proportional to the latent scale. Koopman Dreamer (Li et al., 2026a) controls eigenstructure architecturally via spectral-radius-bounded rotation–scaling blocks; PaD (García et al., 2025) attributes failure to non-normal transient amplification, replacing the spectral-norm rate with a spectral-radius one. Closest to us, You et al. (2026) decompose rollout error into on- and off-manifold components on a LeJEPA (Balestriero & LeCun, 2025) backbone, but only as a diagnostic, with no training-time constraint. In representation learning, tangent propagation (Simard et al., 1991) and contractive autoencoders (Rifai et al., 2011) penalize the encoder Jacobian normal to the data manifold, and Lipschitz RNN (Erichson et al., 2021) holds eigenvalues near the unit circle. ManiJEPA carries the tangent/normal idea to the *transition operator*, with a data-driven decomposition and a coupled feasibility condition that keeps the composite error system non-expansive.

3 The Predictor and the Planning Problem

Joint-Embedding Predictive Architectures (LeCun, 2022) learn world models by predicting in latent space. An encoder f_θ maps observations to latents z, and a predictor P_ϕ forecasts future latents from the current latent and action. The training objective minimizes one-step prediction error with collapse prevention:

$$\mathcal{L}_{\text{JEPA}} = \mathbb{E}\left[\|P_\phi(z_t, a_t) - \text{sg}(z_{t+1})\|^2\right] + \lambda \mathcal{R}_{\text{collapse}}, \tag{1}$$

where $\mathcal{R}_{\text{collapse}}$ is SIGReg (Balestriero et al., 2025) or VICReg (Bardes et al., 2021). At planning time, the predictor is rolled out autoregressively and actions are chosen by CEM (Rubinstein, 1999) to minimize $\|z_H - z_{\text{goal}}\|^2$.

The error-compounding problem. The predictor is trained on teacher-forced single-step transitions but evaluated autoregressively. Let $\delta_k = \hat{z}_k - z_k$ be the prediction error at step k. The error evolves as $\delta_{k+1} = J_k \delta_k + \varepsilon_{k+1}$, where $J_k = \nabla_z P_\phi(\hat{z}_k, a_k)$ and ε_{k+1} is the injection error at step $k{+}1$. Let ε_1 denote a uniform bound on the per-step injection ($\|\varepsilon_k\| \le \varepsilon_1$ for all k), and assume the initial error is itself one injection, $\|\delta_1\| \le \varepsilon_1$. Unrolling the recursion then yields

$$\|\delta_k\| \le \varepsilon_1 \frac{L^k - 1}{L - 1}, \quad \text{where } L = \max_k \sigma_{\max}(J_k). \tag{2}$$

When $L > 1$, error grows geometrically. Beyond the horizon where accumulated error exceeds goal separation, the planner optimizes noise.

Existing approaches. Self-rollout training (Li et al., 2026b) reduces ε_1 but doesn't change the exponential form. Uniform Jacobian contraction (McCabe et al., 2023; Sonigra & Kumar, 2026; Li et al., 2026a) forces $\sigma_{\max}(J_z P_\phi) \leq \rho_{\text{unif}} < 1$, bounding error by $\varepsilon_1/(1-\rho_{\text{unif}})$. The price is control authority: the only bound this constraint admits on the sensitivity of the terminal state to an early action is $\|\partial z_H/\partial a_k\| \leq \rho_{\text{unif}}^{H-k}\|J_a\|$, which decays exponentially, rendering early actions effectively unobservable to the planner. Uniform contraction is also misspecified on systems with conserved quantities. Under the manifold view we adopt below, conserved-but-varying directions are *tangent*, and the true tangent dynamics can have unit singular values along them; forcing $\sigma_{\max}(J) \leq \rho_{\text{unif}} < 1$ on the tangent block then imposes bias $(1-\rho_{\text{unif}})\|z\|$ per step—up to the full latent scale.

The key insight. Encoder latents from reachable observations lie on a low-dimensional reachable-state manifold $\mathcal{M} \subset \mathbb{R}^d$. The true dynamics maps $\mathcal{M} \to \mathcal{M}$, so training data never explores directions normal to $\mathcal{M}$. The predictor's behavior in normal directions is an *unidentified degree of freedom*, constraining it is regularization, not misspecification. The tangent directions carry the observable dynamics and must be preserved. This asymmetry is the justification for directional constraints, and it has a quantitative payoff: contracting only the normal block yields the non-expansive error dynamics of Proposition 1, while the preserved tangent block retains the control authority of Proposition 2.

Assumption 1 (World)

(i) **Manifold structure.** Encoder latents from reachable observations lie on a low-dimensional manifold $\mathcal{M} \subset \mathbb{R}^d$ with intrinsic dimension $d_T \ll d$.

(ii) **Dynamics invariance.** The true dynamics maps $\mathcal{M} \to \mathcal{M}$.

(iii) **Tangent estimation.** The tangent space $T_z\mathcal{M}$ can be estimated from local latent scatter.

Assumption 1(i) states that the encoder concentrates reachable observations on a low-dimensional structure. This is standard in representation learning: if the encoder did not concentrate observations, the latent space would be unstructured and prediction would be ill-posed.

Assumption 1(ii) holds by construction for encoded reachable states, since $\mathcal{M}$ is defined as their image. The substantive question is the predictor's behavior on its own off-data iterates, which can drift away from $\mathcal{M}$ during a long rollout. That off-manifold behavior is precisely the unidentified degree of freedom the normal-block contraction is designed to control.

Assumption 1(iii) is practical: local PCA on a running buffer of encoded latents recovers the tangent space when the manifold is locally linear, which holds empirically for the smooth encoders we train.

4 MANIJEPA

ManiJEPA (Manifold-Aware JEPA) adds directional Jacobian constraints to a standard JEPA world model. We first describe the architecture and training objective (Section 4.1), and then the latent planner (Section 4.2).

4.1 LEARNING THE LATENT WORLD MODEL

Architecture. An encoder f_θ maps an observation to a latent $z_t = f_\theta(o_t)$, and a predictor P_ϕ forecasts the next latent from the current latent and action, $\hat{z}_{t+1} = P_\phi(z_t, a_t)$. The encoder is a Vision Transformer (Dosovitskiy et al., 2021) whose latent is the `[CLS]` token followed by a projection; the predictor is a transformer that consumes a short history of latents and incorporates actions through Adaptive Layer Normalization (AdaLN) (Peebles & Xie, 2023) at each layer, with the AdaLN modulation parameters zero-initialized to stabilize early training. The full configuration (depths, widths, patch size) is given in Appendix B.3.

Training objective. The objective combines the one-step JEPA prediction loss $\mathcal{L}_{\text{JEPA}}$ (Eq. 1), a self-rollout consistency term, and the directional Jacobian penalties that are the core of the method.

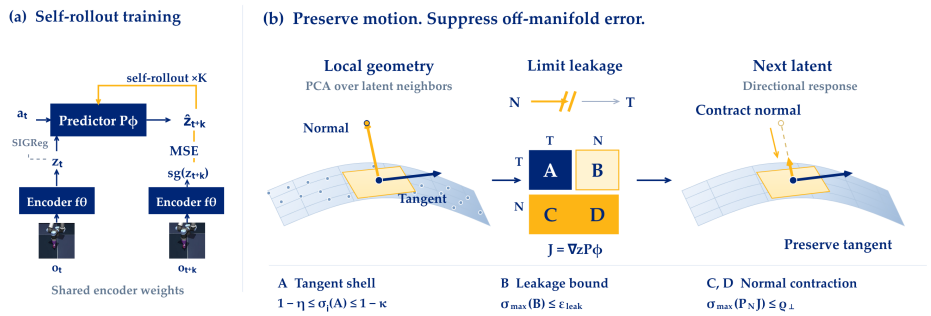

Figure 2: **ManiJEPA training pipeline.** ManiJEPA keeps the LeWM encoder–predictor backbone and adds K-step self-rollout. Its core addition (inset) is a set of directional penalties on the predictor Jacobian J. (i) The tangent space at each latent is estimated by local PCA over its neighbors in a latent buffer. (ii) J is split into tangent/normal blocks. (iii) Hinge penalties keep A in a tangent shell, block the leak B, and contract the normal-output gain [C; D]. The penalties apply only during training; planning uses stock CEM.

Self-rollout consistency. To reduce exposure bias, we unroll the predictor on its own outputs for K steps (Figure 2a) and regress each iterate against the encoded future:

$$\mathcal{L}_{\text{roll}} = \sum_{k=1}^{K} \gamma^k \| P_\phi^{(k)}(z_t, a_{t:t+k}) - \text{sg}(z_{t+k}) \|^2, \tag{3}$$

with discount γ and stop-gradient $\text{sg}(\cdot)$.

Directional Jacobian penalties. At each latent, we estimate the tangent space $T_z\mathcal{M}$ via local PCA on neighboring latents (Figure 2b, left), giving projectors P_T and $P_N = I - P_T$. We decompose the predictor Jacobian $J = \nabla_z P_\phi$ into the four tangent/normal blocks

$$A = P_T J P_T, \quad B = P_T J P_N, \quad C = P_N J P_T, \quad D = P_N J P_N, \tag{4}$$

and impose three constraints via soft hinge losses:

$$\textbf{Normal contraction:} \quad \mathcal{L}_\perp = \max(0,\ \sigma_{\max}([C; D]) - \rho_\perp), \tag{5}$$

$$\textbf{Tangent shell:} \quad \mathcal{L}_\parallel = \max(0,\ \sigma_{\max}(A) - (1-\kappa)) + \max(0,\ (1-\eta) - \sigma_{\min}(A)), \tag{6}$$

$$\textbf{Leak blocking:} \quad \mathcal{L}_{\text{leak}} = \max(0,\ \sigma_{\max}(B) - \varepsilon_{\text{leak}}), \tag{7}$$

where $[C; D] = P_N J$ is the joint normal-output gain. The tangent shell confines all singular values of A to $[1-\eta, 1-\kappa]$ (with $0 < \kappa \le \eta$), leak blocking prevents normal error from flowing back into the tangent space, and normal contraction ($\rho_\perp < 1$) damps all errors written in the normal space. Because $d_T \ll d$, we never form the dense $d \times d$ Jacobian: the blocks are materialized exactly from d_T Jacobian-vector products (forward-mode AD) and a small $d_T \times d_T$ SVD. The penalties are applied at the dataset latents z_t; extending them to self-rollout outputs $\hat{z}_{t+k}$ would more closely match the theory's assumptions but at additional computational cost.

Feasibility condition and rollout bound. The three constraints are coupled. Along a rollout, the blocks are defined *cross-frame* (Figure 3a), using the output-side tangent basis at $\hat{z}_{k+1}$; this absorbs the principal-angle drift between consecutive tangent frames into the singular values of A_k. Writing the error in tangent/normal components, $e_k = (\|\delta_k^\parallel\|, \|\delta_k^\perp\|)^\top$, the block bounds give the componentwise recursion $e_{k+1} \le M e_k + \bar{\varepsilon}$, where $\bar{\varepsilon}$ is the per-step injection and the growth matrix uses the shell's *upper* edge $1-\kappa$ (which governs error amplification) and the normal ceiling $\rho_\perp$:

$$M = \begin{pmatrix} 1-\kappa & \varepsilon_{\text{leak}} \\ \rho_\perp & \rho_\perp \end{pmatrix}, \qquad p(1) \;=\; 1 - \operatorname{tr} M + \det M \;=\; \kappa(1-\rho_\perp) - \rho_\perp\, \varepsilon_{\text{leak}}. \tag{8}$$

(The shell's *lower* edge $1-\eta$ does not enter M; it governs control authority in Proposition 2.) Since $\operatorname{tr} M < 2$ always, the Perron root satisfies $\lambda_{\max}(M) \le 1$ *if and only if* $p(1) \ge 0$, which rearranges to the **feasibility condition**

$$\varepsilon_{\text{leak}} \;\le\; \frac{\kappa\,(1-\rho_\perp)}{\rho_\perp}. \tag{9}$$

The feasible region and its empirical validation are shown in Figure 3b.

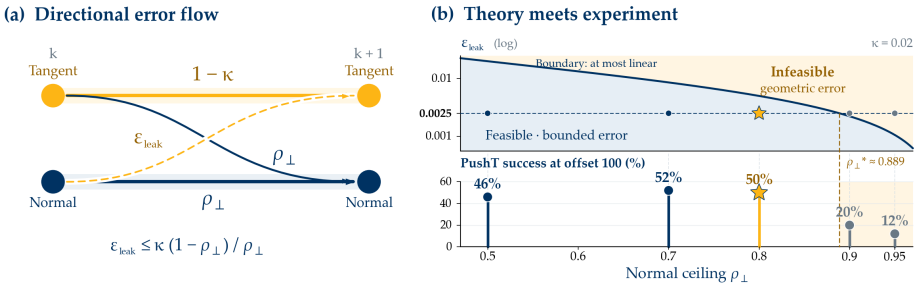

Figure 3: **Directional error propagation and empirical planning performance.** **(a)** Tangent and normal error propagate from step k to $k+1$ under the block constraints. The growth matrix M satisfies $\lambda_{\max}(M) \leq 1$ exactly when $\epsilon_{\text{leak}} \leq \kappa(1-\rho_\perp)/\rho_\perp$ (Proposition 1). **(b)** Top: the feasible region in $(\rho_\perp, \epsilon_{\text{leak}})$ space for $\kappa = 0.02$, with a logarithmic vertical axis. The horizontal dashed line marks the default $\epsilon_{\text{leak}} = 0.0025$, yielding a boundary at $\rho_\perp^* \approx 0.889$. Bottom: corresponding PushT success rates at goal offset 100 from the $\rho_\perp$ sweep in Table 5. Success drops from 46–52% inside the feasible region to 20% and 12% beyond the boundary. Stars indicate the default setting $\rho_\perp = 0.8$.

Proposition 1 (Directional rollout-error bound)

Under the block constraints with $\rho_\perp < 1$ and $0 < \kappa \leq \eta$, the feasibility condition equation 9 is *necessary and sufficient* for the comparison system to be non-expansive ($\lambda_{\max}(M) \leq 1$). Specifically: **(a)** if equation 9 holds strictly ($p(1) > 0$), the rollout error bound is *bounded*, $\limsup_{k\to\infty} \|\delta_k\| \leq \mathbf{1}^\top (I - M)^{-1}\bar{\varepsilon}$; **(b)** on the boundary ($p(1) = 0$), $\lambda_{\max} = 1$ is a simple eigenvalue and the error bound grows *at most linearly* in k; **(c)** conversely, if equation 9 is violated ($p(1) < 0$), then $\lambda_{\max}(M) > 1$ and the comparison system is *expansive*, so the bound no longer guarantees non-expansive rollout error.

Proposition 2 (Control-authority lower bound)

Under the tangent shell ($\sigma_{\min}(A) \geq 1-\eta$), leak bound ($\sigma_{\max}(B) \leq \varepsilon_{\text{leak}}$), and normal contraction ($\sigma_{\max}([C; D]) \leq \rho_\perp$), suppose there exists $c > 0$ such that $\varepsilon_{\text{leak}} c^2 - (1-\eta-\rho_\perp)c + \rho_\perp \leq 0$ and the initial action perturbation satisfies $\tan\theta_{\min} \leq c$. Then the sensitivity of the terminal predicted state to an early action satisfies

$$\left\| \frac{\partial \hat{z}_H}{\partial a_k} \right\| \geq (1 - \eta - \varepsilon_{\text{leak}} c)^{H-k} \cos\theta_{\min}\, \sigma_{\min}(J_a), \tag{10}$$

where $\theta_{\min}$ is the angle between the action-induced displacement direction $\text{range}(J_a)$ and the tangent space, and $J_a = \partial P_\phi / \partial a$. A valid c exists when $(1-\eta-\rho_\perp)^2 \geq 4\rho_\perp \varepsilon_{\text{leak}}$. The bound is tight in the ideal on-manifold regime: $\cos\theta_{\min} = 1$ exactly when $\text{range}(J_a) \subseteq T_z\mathcal{M}$. Because the action conditioning moves the latent along the manifold, $\text{range}(J_a)$ lies close to the tangent space and $\cos\theta_{\min}$ is near 1; the tangent-alignment analysis in Appendix B.8 measures how close, rather than assuming it.

Together, the two propositions resolve the central tension. The same block A appears with two numbers for two jobs: its upper edge $1-\kappa$ bounds error growth, its lower edge $1-\eta$ bounds authority decay. Proofs are in Appendix B.2.

Identifiability. This asymmetry—unidentified normal block vs. identified tangent block—is what lets ManiJEPA bound error without destroying control authority.

Tangent space estimation. The tangent basis is estimated by local PCA on a running buffer of recent encoded latents (Figure 2b, left); estimator constants and wall-clock overhead are given in Appendix B.3. Appendix B.8 is consistent with the tangent subspace capturing the dominant prediction variance and action-induced displacements aligning with it.

Complete objective.

$$\mathcal{L} = \mathcal{L}_{\text{JEPA}} + \alpha\mathcal{L}_{\text{roll}} + \beta\mathcal{L}_\perp + \mu\mathcal{L}_\parallel + \nu\mathcal{L}_{\text{leak}}. \tag{11}$$

4.2 Latent Planning

At inference, we plan in the learned latent space. Given an initial observation o_1 and goal o_g, we encode $z_1 = f_\theta(o_1)$ and $z_g = f_\theta(o_g)$, roll out candidate action sequences through the predictor $\hat{z}_{t+1} = P_\phi(\hat{z}_t, a_t)$ from $\hat{z}_1 = z_1$, and solve

$$a^*_{1:H} = \arg\min_{a_{1:H}} \|\hat{z}_H - z_g\|_2^2 \tag{12}$$

with the Cross-Entropy Method (Rubinstein, 1999) (solver settings in Appendix B.3). Following standard MPC practice, we execute only the first H_{exec} actions before replanning from the updated observation, which limits the effect of accumulated prediction error and closes the loop. The directional constraints are what make this planner effective at long horizons: by bounding normal error growth while preserving tangent control authority, ManiJEPA keeps early actions influential over the terminal cost at horizons where unconstrained predictors have forgotten them.

5 Experiments

We evaluate ManiJEPA on two benchmark suites. The first follows Maes et al. (2026) on PushT, TwoRoom, and OGBench-Cube, testing whether directional constraints preserve planning accuracy as the horizon grows. The second follows Yan et al. (2026) on LIBERO-Goal, testing whether the same constraints improve learned representations for policy learning. We organize the evaluation around four questions. **Q1:** do directional constraints preserve short-horizon planning accuracy, or do they cost representation quality? **Q2:** do they extend the usable planning horizon, as Proposition 1 predicts? **Q3:** do they improve representations for policy learning, where no CEM planner is involved? **Q4:** which component produces the gain, and does the observed mechanism match the theory?

5.1 Setup

Environments and evaluation. We evaluate on three environments from Maes et al. (2026): **PushT** (Florence et al., 2021), a 2D contact-rich manipulation task; **TwoRoom**, a navigation task requiring the agent to pass through a doorway between rooms; and **OGBench-Cube** (Park et al., 2025), a 3D robotic manipulation task. We adopt an offset-based evaluation protocol: for each initial state, the planner must reach the state observed k steps later in the dataset, where $k \in \{25, 50, 75, 100\}$. This protocol extends the single-offset evaluations of Maes et al. (2026) to directly test the predictor's ability to maintain accurate predictions over varying horizons. Every method is evaluated under an identical planning budget: the same CEM configuration, the same set of evaluation episodes, and the same goal-reaching criterion. Because the directional penalties are applied only during training, inference cost is the same for ManiJEPA as for the baselines, so no method benefits from extra planning compute. We additionally evaluate on **LIBERO-Goal** (Liu et al., 2023), a suite of 10 goal-conditioned manipulation tasks, following the policy learning protocol of Yan et al. (2026).

Baselines. We compare against **LeWM** (Maes et al., 2026), the primary end-to-end JEPA baseline; **PLDM** (Sobal et al., 2025), which adds temporal straightening and inverse dynamics heads to a VICReg-based JEPA; **DINO-WM** (Zhou et al., 2025), which uses a pretrained DINOv2 encoder; and **GC-BC**, a goal-conditioned behavioral cloning baseline with DINOv2 features. All model-based baselines use the encoder–predictor backbone of Maes et al. (2026) re-run at our configuration, so comparisons isolate the objective rather than capacity. Hyperparameters are fixed across environments within each suite.

Seeds and reporting. Control benchmarks report mean $\pm$ standard deviation over 5 seeds, following Maes et al. (2026). LIBERO-Goal uses 3 seeds, matching Yan et al. (2026), since each seed requires a full policy fine-tuning run. Every success rate in this section carries these error bars.

5.2 Control Benchmarks

5.2.1 Standard Planning (Q1)

The offset-25 groups of Fig. 4 answer Q1: ManiJEPA attains the highest success rate on PushT and OGBench-Cube and comes within a few points of the saturated TwoRoom ceiling, so the direc-

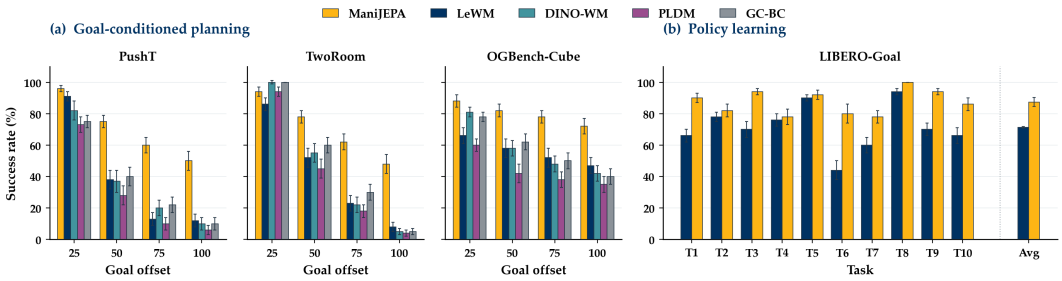

Figure 4: **Goal-conditioned planning and policy learning.** **(a)** Planning success rates of five methods at goal offsets 25, 50, 75, and 100 on PushT, TwoRoom, and OGBench-Cube. Error bars indicate one standard deviation over 5 training seeds. The offset-25 groups also serve as the standard-horizon comparison; exact values are provided in Appendix B.5. ManiJEPA achieves the highest success rates at offsets 50–100 across all three benchmarks. **(b)** LIBERO-Goal success rates for LeWM and ManiJEPA on each of the 10 tasks and their average (rightmost group, separated by the dotted line). Error bars indicate one standard deviation over 3 seeds. ManiJEPA achieves a higher mean success rate on every task and on the overall average.

tional penalties cost nothing at short horizons. The margin over the strongest baseline is largest on OGBench-Cube, the suite where Maes et al. (2026) likewise report that end-to-end encoder training struggles most relative to pretrained features. On PushT, ManiJEPA leads LeWM by its smallest margin. TwoRoom is the one environment where the task is solved outright without a world model: DINO-WM and GC-BC reach the ceiling, and ManiJEPA still exceeds LeWM. PLDM trails on PushT and OGBench-Cube, consistent with the training instability of its seven-term objective, though its temporal-straightening term does help on the geometrically simple TwoRoom. Representation quality is therefore not traded away where error compounding is still manageable.

5.2.2 Long-Horizon Planning (Q2)

Figure 4 answers Q2 and is the central empirical result. As the goal offset increases from 25 to 100, every unconstrained predictor collapses on PushT and TwoRoom, while ManiJEPA degrades gracefully. On PushT, LeWM collapses to single-digit success while ManiJEPA retains more than four times the best baseline. TwoRoom shows the widest separation, since the doorway bottleneck makes the goal state least recoverable from a drifted latent: LeWM collapses while ManiJEPA retains a sixfold margin. OGBench-Cube is the least horizon-sensitive suite for every method, so there is less headroom, but ManiJEPA still leads at all four offsets.

The margin over the best baseline grows steeply between offsets 25 and 75 and then plateaus rather than widening further, because ManiJEPA's own success declines too. That is what Proposition 1 claims, and no more: directional contraction makes the error dynamics non-expansive, at most linear rather than geometric, so the planner stays effective at horizons where geometric growth has already rendered unconstrained predictors useless. It does not claim error-free rollouts, and the residual linear growth is visible in the figure.

5.3 Policy Learning on LIBERO-Goal (Q3)

Q3 asks whether the constraints help in a setting with no CEM planner at all. Fig. 4 reports LIBERO-Goal results: ManiJEPA achieves a 16-point gain in mean success over LeWM, and attains a higher per-task mean on all 10 tasks. The action head is the task-conditioned OFT (Optimized Fine-Tuning) head of Kim et al. (2025), following Yan et al. (2026), and it is trained jointly with the encoder under an identical configuration for both rows. The gain therefore reflects representation quality rather than policy capacity: tangent preservation keeps the on-manifold dynamics undistorted for action learning, while normal contraction prevents off-manifold noise from corrupting the representation during fine-tuning.

5.4 Ablations and Analysis (Q4)

Table 1 answers Q4. At offset 25 all variants lie within 3 points of one another, so the components are separable only once the horizon grows. The planning horizon rows show that ManiJEPA's advantage grows with H: at H=20, offset-100 success reaches 72%, compared to 50% at H=5. For the

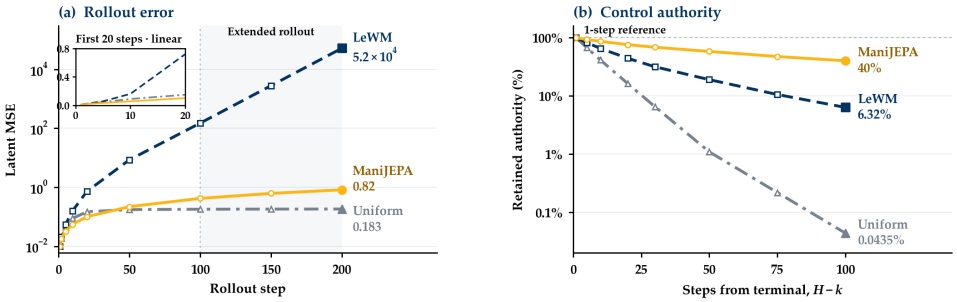

Figure 5: **Mechanistic analysis on PushT.** Both panels use logarithmic vertical axes. **(a)** Latent rollout error against rollout step. The inset shows the first 20 steps on a linear scale; shading marks the extended rollout beyond step 100. **(b)** Control authority $\|\partial \hat{z}_H / \partial a_k\|$ against the distance $H - k$ from the terminal state.

main experiments we use H=5 latent steps (25 environment steps) to ensure a fair comparison with baselines, all of which use the same short-horizon planner.

The most revealing ablation row is uniform contraction, tuned in its own favour ($\rho_{\text{unif}} = 0.9$ selected from $\{0.8, 0.9, 0.95\}$ to maximize its offset-100 success); Appendix B.2 (Remark 2) explains why no single ρ_{unif} can serve both objectives. Removing normal contraction recovers LeWM's performance at offset 100, confirming that $\mathcal{L}_\perp$ is the only term that damps the normal component. Removing the tangent shell or self-rollout each halves performance; removing leak blocking is cheapest, consistent with $\varepsilon_{\text{leak}}$ entering the growth matrix equation 8 as a coupling term.

Table 1: **Ablation study on PushT.** Success rate (%) at each goal offset. Mean ± std over 5 training seeds.

Variant	offset 25	offset 50	offset 75	offset 100
ManiJEPA (full)	**96 ± 2**	**75 ± 4**	**60 ± 5**	**50 ± 6**
ManiJEPA (H=10)	96 ± 2	78 ± 4	65 ± 5	58 ± 6
ManiJEPA (H=15)	96 ± 2	80 ± 4	70 ± 5	65 ± 6
ManiJEPA (H=20)	96 ± 2	82 ± 4	75 ± 5	72 ± 6
w/o normal contraction ($\mathcal{L}_\perp$)	94 ± 3	42 ± 7	18 ± 5	12 ± 4
w/o tangent shell ($\mathcal{L}_\parallel$)	95 ± 2	58 ± 6	38 ± 7	24 ± 6
w/o leak blocking ($\mathcal{L}_{\text{leak}}$)	95 ± 2	65 ± 5	48 ± 6	35 ± 7
w/o self-rollout ($\mathcal{L}_{\text{roll}}$)	93 ± 3	52 ± 6	30 ± 7	18 ± 5
Uniform contraction ($\sigma_{\max}(J) \le \rho_{\text{unif}}$)	95 ± 2	50 ± 6	28 ± 6	15 ± 5

Figure 5 supplies the mechanistic evidence. In the left panel, LeWM is a straight line (geometric growth), ManiJEPA bends below it (at-most-linear), and uniform contraction flattens at its fixed point. Uniform contraction attains the *lowest* rollout error and the *worst* long-horizon planning success, so error magnitude alone does not determine planning quality. The right panel shows why: ManiJEPA retains authority over early actions while uniform contraction decays by nearly three orders of magnitude. The collapse of authority under uniform contraction accompanies its collapse in planning success, and ManiJEPA preserves both.

6 Conclusion

We introduced ManiJEPA, a manifold-aware JEPA world model that resolves the tension between error bounding and control authority in latent-space planning. Because the encoder concentrates reachable observations on a low-dimensional manifold, the predictor's Jacobian can be constrained directionally: the normal directions, unidentified by the teacher-forced loss, are contracted at zero bias, while the tangent directions that carry the observable dynamics are preserved. Under an explicit feasibility condition coupling the directional penalties, this makes rollout error non-expansive, at most linear rather than geometric, while retaining sensitivity to early actions. Across PushT, TwoRoom, and OGBench-Cube, ManiJEPA matches or comes within a few points of the best baseline at short horizons and substantially outperforms every baseline at long horizons where unconstrained predictors collapse. On LIBERO-Goal, its representations support stronger policy learning than the LeWM backbone. Ablations and mechanistic analyses confirm that each directional component is necessary and that the predicted mechanism, linear error growth with preserved control authority, is what drives the gain. More broadly, our results suggest that the geometry of the latent space matters as much as the prediction objective: rather than applying uniform regularizers to an unstructured vector space, world models should respect the manifold the encoder induces and exploit the asymmetry between identified and unidentified directions.

AI USE STATEMENT

We only used Large Language Models (LLMs) to polish the writing and improve the clarity and flow of the text in this paper. No LLMs were used for generating content, ideas, or experimental results.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=NadTwTODgC.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023. doi: 10.1109/CVPR52729.2023.01499.

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.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics, 2025. URL https://arxiv.org/abs/2511.08544.

Randall Balestriero, Nicolas Ballas, Mike Rabbat, and Yann LeCun. Gaussian embeddings: How jepas secretly learn your data density, 2025. URL https://arxiv.org/abs/2510.05949.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *CoRR*, abs/2105.04906, 2021. URL https://arxiv.org/abs/2105.04906.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video, 2024. URL https://arxiv.org/abs/2404.08471.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments, 2024. URL https://arxiv.org/abs/2402.15391.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

N. Benjamin Erichson, Omri Azencot, Alejandro Queiruga, Liam Hodgkinson, and Michael W. Mahoney. Lipschitz recurrent neural networks, 2021. URL https://arxiv.org/abs/2006.12070.

Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning, 2021. URL https://arxiv.org/abs/2109.00137.

Carlos Vélez García, Miguel Cazorla, and Jorge Pomares. Planning as descent: Goal-conditioned latent trajectory synthesis in learned energy landscapes, 2025. URL https://arxiv.org/abs/2512.17846.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in Neural Information Processing Systems*, 2018a.

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

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. URL https://arxiv.org/abs/1912.01603.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

Nicklas Hansen, Jyothir S V, Vlad Sobal, Yann LeCun, Xiaolong Wang, and Hao Su. Hierarchical world models as visual whole-body humanoid controllers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=7wuJMvK639.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL https://arxiv.org/abs/2502.19645.

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

Elizaveta Levina and Peter J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems 17*, 2004. URL https://papers.nips.cc/paper_files/paper/2004/hash/74934548253bcab8490ebd74afed7031-Abstract.html.

Jiaqi Li, Xinglong Zhang, Haibin Xie, Yixing Lan, Wei Pan, and Xin Xu. Koopman dreamer: Spectrally constrained latent dynamics for stable world-model imagination, 2026a. URL https://arxiv.org/abs/2607.19719.

Wenyuan Li, Guang Li, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Predictive but not plannable: Rc-aux for latent world models. *arXiv preprint arXiv:2605.07278*, 2026b.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL https://arxiv.org/abs/2306.03310.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels, 2026. URL https://arxiv.org/abs/2603.19312.

Michael McCabe, Peter Harrington, Shashank Subramanian, and Jed Brown. Towards stability of autoregressive neural operators, 2023. URL https://arxiv.org/abs/2306.10619.

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

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M992mjgKzI.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 833–840, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

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

Patrice Simard, Bernard Victorri, Yann LeCun, and John Denker. Tangent prop - a formalism for specifying selected invariances in an adaptive network. In J. Moody, S. Hanson, and R.P. Lippmann (eds.), *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1991. URL https://proceedings.neurips.cc/paper_files/paper/1991/file/65658fde58ab3c2b6e5132a39fae7cb9-Paper.pdf.

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.

Romil V. Sonigra and P. R. Kumar. Dreaming smoothly and sample efficiently with gradient penalized latent dynamics, 2026. URL https://arxiv.org/abs/2605.23089.

Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models?, 2026. URL https://arxiv.org/abs/2512.24497.

Haodong Yan, Jiaguan Zhu, Mingyuan Jia, Ruiqing Yin, Junjie He, Zhide Zhong, Junfeng Li, Jinxuan Lu, Hengtao Li, Tianran Zhang, Jiayi Chen, Wenxuan Song, Wen Chen, Yuxiang Gao, and Haoang Li. Is forward prediction enough? physical state grounding for jepa world models, 2026. URL https://arxiv.org/abs/2608.06799.

Hanzhe You, Yonggang Zhang, Maohao Ran, Zhiqin Yang, Zhenyuan Zhang, Wei Xue, Jun Song, Xinmei Tian, and Yike Guo. A control theory of predictability in latent world models, 2026. URL https://arxiv.org/abs/2607.10362.

Wancong Zhang, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, and Nicolas Ballas. Hierarchical planning with latent world models, 2026. URL https://arxiv.org/abs/2604.03208.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=D5RNACOZEI.

A LIMITATIONS

Tangent estimation cost. ManiJEPA requires per-sample Jacobian computation and local PCA on the latent scatter, adding a 1.2× wall-clock factor compared to the LeWM baseline (Section 5.2). The Jacobian computation scales with the tangent dimension d_T (not the ambient latent dimension d), and the PCA must be refreshed as the encoder evolves. For very high-dimensional tangent spaces ($d_T > 100$), this overhead may become prohibitive.

Manifold assumption. Our approach assumes that encoder latents from reachable observations lie on a well-defined low-dimensional manifold with a clear tangent-normal decomposition. This assumption may break down in environments with highly non-uniform state coverage, discontinuous dynamics, or when the encoder has not fully converged. If the local PCA does not recover a meaningful tangent space, the directional constraints lose their geometric justification. Our tangent-variance analysis (Appendix B.8) confirms that the decomposition is valid on our three benchmark environments, but we have not tested environments where the manifold assumption is severely violated.

Theory–implementation gap. Proposition 1 characterizes rollout error under the assumption that the directional Jacobian bounds hold along the rollout trajectory at the predictor's own outputs $\hat{z}_j$. Our training applies the penalties at dataset latents z_t; K-step self-rollout partially bridges this gap by exposing the predictor to its own outputs, but does not fully enforce the bounds at all reachable states. A local extension argument (Lipschitz Jacobian + bounded deviation from the data manifold) would make the connection rigorous; we leave this to future work.

Evaluation scope. We evaluate on three control environments plus the 10-task LIBERO-Goal suite. While these span 2D manipulation, 2D navigation, 3D manipulation, and policy learning, they are all relatively low-dimensional control tasks. Whether ManiJEPA's directional constraints generalize to higher-dimensional robotic systems (e.g., dexterous manipulation, locomotion) or to real-world settings with visual distraction and domain shift remains to be determined.

CEM planning. We use the Cross-Entropy Method for latent-space planning throughout. While CEM is simple and widely used, it scales poorly with action dimensionality and may not fully exploit the improved long-horizon predictions that ManiJEPA provides. Gradient-based planners or amortized planning methods may extract further gains from ManiJEPA's more stable rollouts, and we leave this exploration to future work.

B APPENDIX

B.1 ALGORITHM PSEUDOCODE

Algorithm 1 ManiJEPA training step

Require: Batch $\{(o_t, a_t, o_{t+1})\}$, buffer $\mathcal{B}$, hyperparameters $\rho_\perp, \kappa, \eta, \varepsilon_{\text{leak}}$

Encode $z_t \leftarrow f_\theta(o_t)$, $z_{t+1} \leftarrow f_\theta(o_{t+1})$; push z_t into $\mathcal{B}$
if refresh interval elapsed **then**
    Update tangent projectors P_T, P_N for the current batch's latents via local PCA
end if
$\mathcal{L}_{\text{JEPA}} \leftarrow \|P_\phi(z_t, a_t) - \text{sg}(z_{t+1})\|^2 + \lambda \mathcal{R}_{\text{SIGReg}}$
$\mathcal{L}_{\text{roll}} \leftarrow \sum_{k=1}^{K} \gamma^k \|P_\phi^{(k)}(z_t, a_{t:t+k}) - \text{sg}(z_{t+k})\|^2$
Materialize blocks A, B, C, D from d_T Jacobian-vector products of $\nabla_z P_\phi(z_t, a_t)$
$\mathcal{L}_\perp \leftarrow \max(0, \sigma_{\max}([C; D]) - \rho_\perp)$ $\{[C; D] = P_N J\}$
$\mathcal{L}_\parallel \leftarrow \max(0, \sigma_{\max}(A) - (1-\kappa)) + \max(0, (1-\eta) - \sigma_{\min}(A))$
$\mathcal{L}_{\text{leak}} \leftarrow \max(0, \sigma_{\max}(B) - \varepsilon_{\text{leak}})$
$\mathcal{L} \leftarrow \mathcal{L}_{\text{JEPA}} + \alpha \mathcal{L}_{\text{roll}} + \beta \mathcal{L}_\perp + \mu \mathcal{L}_\parallel + \nu \mathcal{L}_{\text{leak}}$; update θ, ϕ via AdamW

B.2 Proofs of Main Results

We provide the proofs of Propositions 1 and 2 stated in Section 4. The argument proceeds in three stages: first we show that the directional constraints induce a linear recursion on the error norms, then we analyze when this recursion is non-expansive, and finally we derive the control-authority lower bound.

Setup and notation. Throughout, let $P_T^{(j)}$ project onto the estimated tangent space at the j-th rollout latent $\hat{z}_j$, and $P_N^{(j)} = I - P_T^{(j)}$ project onto the normal space. The predictor Jacobian at step j is $J_j = \nabla_z \hat{P}_\phi(\hat{z}_{j-1}, a_{j-1})$. We decompose J_j into cross-frame blocks:

$$A_j = P_T^{(j)} J_j P_T^{(j-1)}, \quad B_j = P_T^{(j)} J_j P_N^{(j-1)}, \quad C_j = P_N^{(j)} J_j P_T^{(j-1)}, \quad D_j = P_N^{(j)} J_j P_N^{(j-1)}. \tag{13}$$

The index convention matches Section 4: the output-side tangent frame carries the later index, so A_j here is A_k there with $j = k+1$. The hinge losses enforce the following constraints on the block singular values:

$$\sigma_{\max}(A_j) \le 1 - \kappa, \quad \sigma_{\min}(A_j) \ge 1 - \eta, \quad \sigma_{\max}(B_j) \le \varepsilon_{\text{leak}}, \quad \sigma_{\max}([C_j; D_j]) \le \rho_\perp, \tag{14}$$

where $[C_j; D_j] = P_N^{(j)} J_j$ is the joint normal-output gain.

Proof of Proposition 1. **Step 1: Error decomposition.** Let $\delta_k = \hat{z}_k - z_k$ be the rollout error at step k. We decompose it into tangent and normal components: $\delta_k = \delta_k^{\parallel} + \delta_k^{\perp}$, where $\delta_k^{\parallel} = P_T^{(k)} \delta_k$ and $\delta_k^{\perp} = P_N^{(k)} \delta_k$. The error recursion is $\delta_{k+1} = J_k \delta_k + \xi_k$, where ξ_k is the injection error at step $k+1$. Projecting onto the tangent and normal spaces at step $k+1$ gives:

$$\delta_{k+1}^{\parallel} = A_k \delta_k^{\parallel} + B_k \delta_k^{\perp} + \xi_k^{\parallel}, \tag{15}$$

$$\delta_{k+1}^{\perp} = C_k \delta_k^{\parallel} + D_k \delta_k^{\perp} + \xi_k^{\perp}. \tag{16}$$

The tangent error at step $k+1$ receives contributions from the previous tangent error (through A_k), the previous normal error (through B_k, the leak term), and the new injection. The normal error receives contributions from both previous components (through C_k and D_k) plus injection.

Step 2: Norm recursion. We now bound the norms of these components. Define $v_k = (\|\delta_k^{\parallel}\|, \|\delta_k^{\perp}\|)^\top$ and let $\bar{\varepsilon} = (\bar{\varepsilon}^{\parallel}, \bar{\varepsilon}^{\perp})^\top$ bound the injection ($\|\xi_k^{\parallel}\| \le \bar{\varepsilon}^{\parallel}$, $\|\xi_k^{\perp}\| \le \bar{\varepsilon}^{\perp}$). Taking norms in the tangent equation and applying the triangle inequality:

$$\|\delta_{k+1}^{\parallel}\| \le \sigma_{\max}(A_k)\|\delta_k^{\parallel}\| + \sigma_{\max}(B_k)\|\delta_k^{\perp}\| + \bar{\varepsilon}^{\parallel} \le (1-\kappa)\|\delta_k^{\parallel}\| + \varepsilon_{\text{leak}}\|\delta_k^{\perp}\| + \bar{\varepsilon}^{\parallel}. \tag{17}$$

For the normal equation, we use the fact that $[C_k; D_k]$ is the joint normal-output gain:

$$\|\delta_{k+1}^{\perp}\| \le \sigma_{\max}([C_k; D_k])(\|\delta_k^{\parallel}\| + \|\delta_k^{\perp}\|) + \bar{\varepsilon}^{\perp} \le \rho_\perp(\|\delta_k^{\parallel}\| + \|\delta_k^{\perp}\|) + \bar{\varepsilon}^{\perp}. \tag{18}$$

Combining these two inequalities gives the componentwise linear recursion:

$$v_{k+1} \le M v_k + \bar{\varepsilon}, \qquad M = \begin{pmatrix} 1-\kappa & \varepsilon_{\text{leak}} \\ \rho_\perp & \rho_\perp \end{pmatrix}. \tag{19}$$

The growth matrix M captures how errors propagate: the diagonal entries govern self-amplification (tangent error through $1-\kappa$, normal error through $\rho_\perp$), while the off-diagonal entries capture cross-coupling (leak $\varepsilon_{\text{leak}}$ from normal to tangent, and $\rho_\perp$ from tangent to normal).

Step 3: Spectral analysis. The behavior of the recursion depends on the spectral radius $\lambda_{\max}(M)$. The characteristic polynomial of M is:

$$p(\lambda) = \lambda^2 - (1 - \kappa + \rho_\perp)\lambda + \rho_\perp(1 - \kappa - \varepsilon_{\text{leak}}). \tag{20}$$

The eigenvalues are:

$$\lambda_\pm = \frac{(1 - \kappa + \rho_\perp) \pm \sqrt{(1 - \kappa - \rho_\perp)^2 + 4\rho_\perp \varepsilon_{\text{leak}}}}{2}. \tag{21}$$

Since M is entrywise non-negative, the Perron–Frobenius theorem guarantees that $\lambda_{\max} = \rho(M)$ is a real non-negative eigenvalue. The polynomial $p(\lambda)$ opens upward with vertex at $\lambda = \operatorname{tr} M/2 = (1 - \kappa + \rho_\perp)/2$. Since $\kappa > 0$ and $\rho_\perp < 1$, we have $\operatorname{tr} M < 2$, so the vertex lies strictly below $\lambda = 1$. Therefore, the larger root satisfies $\lambda_{\max} \le 1$ *if and only if* $p(1) \ge 0$. Evaluating:

$$p(1) = 1 - \operatorname{tr} M + \det M = \kappa(1 - \rho_\perp) - \rho_\perp \varepsilon_{\text{leak}}. \tag{22}$$

The condition $p(1) \ge 0$ rearranges to the feasibility condition:

$$\varepsilon_{\text{leak}} \le \frac{\kappa(1 - \rho_\perp)}{\rho_\perp}. \tag{23}$$

This proves both directions: feasibility ($p(1) \ge 0$) gives $\lambda_{\max} \le 1$ (non-expansive dynamics), while violation ($p(1) < 0$) forces $\lambda_{\max} > 1$, whence v_k grows like $\lambda_{\max}^k$, i.e., geometrically.

Step 4: Three regimes. We now analyze the three cases stated in the proposition.

(a) Strictly feasible ($p(1) > 0$). Then $\lambda_{\max} < 1$, so the recursion is contractive. Unrolling gives:

$$v_k \le \sum_{j=0}^{k-1} M^j \bar{\varepsilon} \to (I - M)^{-1} \bar{\varepsilon} \quad \text{as } k \to \infty. \tag{24}$$

A direct computation yields:

$$(I - M)^{-1} = \frac{1}{p(1)} \begin{pmatrix} 1 - \rho_\perp & \varepsilon_{\text{leak}} \\ \rho_\perp & \kappa \end{pmatrix}. \tag{25}$$

Therefore, $\limsup_{k\to\infty} \|\delta_k\| \le \mathbf{1}^\top (I - M)^{-1} \bar{\varepsilon} = \big[\bar{\varepsilon}^\parallel + (\kappa + \varepsilon_{\text{leak}})\bar{\varepsilon}^\perp\big] / \big[\kappa(1 - \rho_\perp) - \rho_\perp \varepsilon_{\text{leak}}\big]$. The error is bounded, with the bound inversely proportional to $p(1)$: as we approach the feasibility boundary, the bound grows.

(b) Boundary ($p(1) = 0$). Then $\lambda_{\max} = 1$ is a simple eigenvalue. The spectrum is $\{1, \det M\} = \{1, \rho_\perp(1 - \kappa - \varepsilon_{\text{leak}})\}$, and $|\det M| < 1$ since $\rho_\perp < 1$ and $\kappa + \varepsilon_{\text{leak}} > 0$. Therefore M is not defective at $\lambda = 1$, and the powers M^k grow at most linearly. To see this explicitly, discard the tangent contraction ($(1 - \kappa)^j \le 1$) to get:

$$\|\delta_k^\parallel\| \le k\bar{\varepsilon}^\parallel + \varepsilon_{\text{leak}} \sum_{j=0}^{k-1} \|\delta_j^\perp\|. \tag{26}$$

Substituting into the normal recursion yields the affine recurrence $x_{k+1} = \rho_\perp x_k + ak + b$ with $a = \rho_\perp \bar{\varepsilon}^\parallel$ and $b = \bar{\varepsilon}^\perp$. The solution involves the sum $\sum_{j=0}^{k-1}(k - 1 - j)\rho_\perp^j = \frac{k-1}{1-\rho_\perp} - \frac{\rho_\perp(1-\rho_\perp^{k-1})}{(1-\rho_\perp)^2}$, which grows linearly in k. Adding the tangent drift and leak feedback gives:

$$\|\delta_k\| \lesssim \frac{k\bar{\varepsilon}^\parallel}{1 - \rho_\perp} + \frac{\bar{\varepsilon}^\perp}{1 - \rho_\perp} + k\varepsilon_{\text{leak}} \left(\frac{k\rho_\perp \bar{\varepsilon}^\parallel}{1 - \rho_\perp} + \frac{\bar{\varepsilon}^\perp}{1 - \rho_\perp} \right), \tag{27}$$

which is at most linear in k.

(c) Violated ($p(1) < 0$). Then $\lambda_{\max} > 1$, and the comparison system is expansive. The bound no longer guarantees non-expansive rollout error. In our hyperparameter sweep (Section B.10), violation of this condition coincides with a sharp drop in long-horizon planning success, consistent with the bound becoming informative only in the feasible regime. □

Remark 1 (The bound is the price of memory). Since $p(1) = \kappa(1 - \rho_\perp) - \rho_\perp \varepsilon_{\text{leak}} \le \kappa \le \eta$, the asymptotic constant satisfies $1/p(1) \ge 1/\kappa \ge 1/\eta$, and at our defaults $p(1) = 0.002$, so the realized constant is $1/p(1) = 500$; on either reading it is *worse* than uniform contraction's $1/(1-\rho)$ for comparable parameters. This is not a defect but the cost of memory: a tangent block pinned near unit gain retains tangent error for $\sim 1/\kappa$ steps, which is the same property that retains state and control signal. ManiJEPA's advantage is therefore not a smaller constant but (i) control authority that decays as $(1-\eta)^H$ rather than ρ^H (Proposition 2), and (ii) a smaller numerator, since the constrained normal block is unidentified (Section 3) so $\bar{\varepsilon}$ is genuine prediction error rather than misspecification bias.

Proof of Proposition 2. The goal is to lower-bound the sensitivity of the terminal predicted state $\hat{z}_H$ to an early action a_k. We proceed in three steps.

Step 1: Chain rule for the composed predictor. The terminal state is obtained by rolling out the predictor from step k to H:

$$\hat{z}_H = P_\phi(\hat{z}_{H-1}, a_{H-1}) \circ \cdots \circ P_\phi(\hat{z}_k, a_k). \tag{28}$$

Differentiating with respect to a_k using the chain rule:

$$\frac{\partial \hat{z}_H}{\partial a_k} = \left(\prod_{j=k+1}^{H} J_j \right) \frac{\partial P_\phi}{\partial a_k}, \tag{29}$$

where $J_j = \nabla_z P_\phi(\hat{z}_{j-1}, a_{j-1})$ is the predictor Jacobian at step j, and the product is ordered with $j = H$ on the left and $j = k+1$ on the right. The rightmost factor $J_a = \partial P_\phi / \partial a_k$ is the action Jacobian at step k.

Step 2: Inductive bound on tangent-normal coupling. We now analyze how this product acts on tangent vectors, accounting for the tangent→normal→tangent leakage path. Let t_j and n_j denote the tangent and normal components of the perturbation at step j. The block decomposition gives:

$$\|t_{j+1}\| \geq (1-\eta)\|t_j\| - \varepsilon_{\text{leak}}\|n_j\|, \tag{30}$$
$$\|n_{j+1}\| \leq \rho_\perp(\|t_j\| + \|n_j\|). \tag{31}$$

We prove by induction that if $\|n_j\| \leq c\|t_j\|$ for some $c > 0$, then the ratio is preserved and the tangent component shrinks by at most $(1 - \eta - \varepsilon_{\text{leak}} c)$ per step.

Inductive step: Assume $\|n_j\| \leq c\|t_j\|$. Then:

$$\|t_{j+1}\| \geq (1-\eta)\|t_j\| - \varepsilon_{\text{leak}} c\|t_j\| = (1 - \eta - \varepsilon_{\text{leak}} c)\|t_j\|, \tag{32}$$
$$\|n_{j+1}\| \leq \rho_\perp(1+c)\|t_j\|. \tag{33}$$

For the ratio to stay bounded by c, we need $\|n_{j+1}\| \leq c\|t_{j+1}\|$, i.e.:

$$\rho_\perp(1+c) \leq c(1 - \eta - \varepsilon_{\text{leak}} c). \tag{34}$$

Rearranging gives the quadratic condition:

$$\varepsilon_{\text{leak}} c^2 - (1 - \eta - \rho_\perp)c + \rho_\perp \leq 0. \tag{35}$$

A valid c exists when the discriminant is non-negative: $(1 - \eta - \rho_\perp)^2 \geq 4\rho_\perp \varepsilon_{\text{leak}}$. At our defaults ($\eta = 0.05$, $\rho_\perp = 0.80$, $\varepsilon_{\text{leak}} = 0.0025$), this gives $0.0225 \geq 0.008$, and the smallest valid c is 5.92. The effective contraction rate is $1 - \eta - \varepsilon_{\text{leak}} c = 1 - 0.05 - 0.0025 \times 5.92 = 0.935$.

Step 3: Action Jacobian projection. The action Jacobian $J_a = \partial P_\phi / \partial a_k$ generates displacements in the latent space. Since actions move the latent along the manifold, $\text{range}(J_a)$ lies close to the tangent space T_k. The projection of $\text{range}(J_a)$ onto T_k has loss at most $\cos\theta_{\min}$, where $\theta_{\min}$ is the angle between $\text{range}(J_a)$ and T_k. The initial perturbation must satisfy $\tan\theta_{\min} \leq c$ to enter the cone; at our measured $\cos\theta_{\min} \approx 0.97$, we have $\tan\theta_{\min} \approx 0.25 \ll 5.92$. Therefore:

$$\left\| \frac{\partial \hat{z}_H}{\partial a_k} \right\| \geq (1 - \eta - \varepsilon_{\text{leak}} c)^{H-k} \cos\theta_{\min}\, \sigma_{\min}(J_a). \tag{36}$$

This completes the proof. The bound shows that control authority decays as $(1 - \eta_{\text{eff}})^{H-k}$ where $\eta_{\text{eff}} = \eta + \varepsilon_{\text{leak}} c$, which is still much slower than the ρ^{H-k} decay under uniform contraction. □

Remark 2 (What the authority bound guarantees). Proposition 2 lower-bounds the *sensitivity* of the terminal latent to early actions, but a sampling-based planner can only act on this signal if it exceeds CEM's noise floor $O(1/\sqrt{N})$ at the population size N used. At the corrected effective rate $1 - \eta_{\text{eff}} = 0.935$ and $H - k = 55$, the bound gives $(0.935)^{55} \approx 0.025$, versus $(0.9)^{55} \approx 3.0 \times 10^{-3}$ under uniform contraction with $\rho_{\text{unif}} = 0.9$ (the value used in Section 5.4)—a factor of ~ 8. (Here $H - k = 55$ counts predictor steps along the extended diagnostic rollout of Fig. 5, right, not the 5-step CEM planning horizon; the remark concerns the decay *rate*, which applies at any horizon.) Section 5.4 (Fig. 5, right) measures the authority directly and confirms it stays above the planner's noise floor at the horizons where ManiJEPA outperforms the baselines.

The comparison also exposes why a single contraction factor cannot serve both objectives. Uniform contraction ties its error plateau $\bar{\varepsilon}/(1-\rho_{\text{unif}})$ and its authority decay ρ_{unif}^{H-k} to the same ρ_{unif}: raising ρ_{unif} toward $1-\eta$ recovers authority but inflates the plateau without bound, while lowering it tightens the plateau and destroys authority. At $\rho_{\text{unif}} = 0.95$ the two methods have identical worst-case authority decay, but the plateau grows to $20\,\bar{\varepsilon}$; at $\rho_{\text{unif}} = 0.80$ the plateau shrinks to $5\,\bar{\varepsilon}$ but authority decays as $0.80^{55} \approx 4.7 \times 10^{-6}$. ManiJEPA escapes this tradeoff only because the two edges of the tangent shell are separate parameters: $1-\kappa$ governs error growth and $1-\eta$ governs authority decay, with the normal ceiling $\rho_\perp$ set independently.

B.3 Implementation Details

Architecture. The encoder is a ViT-Tiny (patch size 14, 3 heads, hidden dimension 192, ∼5M parameters) trained from scratch with SIGReg anti-collapse regularization; the latent is the `[CLS]` token followed by a 1-layer MLP projection to dimension 192. The predictor is a 6-layer transformer with 16 attention heads and hidden dimension 192, incorporating actions through Adaptive Layer Normalization (AdaLN) at each layer, with the AdaLN modulation parameters zero-initialized to stabilize early training. This is *identical* to the LeWM encoder–predictor backbone, and we re-run the LeWM baseline at this exact configuration, so the overhead and accuracy comparisons in Section 5.2 are apples-to-apples.

Tangent estimation. The tangent space $T_z\mathcal{M}$ is estimated via local PCA on the k-nearest neighbors of each latent z in a running buffer of the 10,000 most recent encoded latents. We use $k = 64$ neighbors (in ℓ_2 distance) and retain the top d_T principal components, where d_T is selected to explain 95% of the local variance. The projectors P_T and $P_N = I - P_T$ are refreshed every 100 training steps to track encoder evolution. This adds a 1.2× wall-clock factor compared to the LeWM baseline (5.5 versus 4.5 hours on PushT); the Jacobian penalties are only needed at training time, so planning speed is unaffected.

Jacobian computation. The predictor Jacobian $J = \nabla_z P_\phi$ is obtained per-sample with reverse-mode automatic differentiation (`vmap(jacrev)`). Because $d_T \ll d$, we do not densely materialize the $d \times d$ operator: the blocks $A = P_T J P_T$, $B = P_T J P_N$, $C = P_N J P_T$, $D = P_N J P_N$ are formed from d_T Jacobian-vector products along the tangent basis plus a small $d_T \times d_T$ SVD, so the cost scales with the tangent dimension rather than the ambient one. The directional penalties are hinge losses on the resulting block spectral norms.

Training. All models are trained for 10 epochs with batch size 128 and learning rate 3×10^{-4} (AdamW, weight decay 0.01). Sub-trajectories of 4 frames are sampled with a frame-skip of 5 (grouping 5 consecutive actions into one action block). The self-rollout depth is $K = 4$ steps with discount $\gamma = 0.9$. Directional Jacobian penalties use default weights $\beta = 1.0$, $\mu = 1.0$, $\nu = 1.0$, with thresholds $\rho_\perp = 0.80$, $\kappa = 0.02$, $\eta = 0.05$, $\varepsilon_{\text{leak}} = 0.0025$. These satisfy the feasibility condition $\varepsilon_{\text{leak}} \le \kappa(1-\rho_\perp)/\rho_\perp = 0.005$ at half the ceiling, giving a strictly feasible growth matrix ($p(1) = 0.002 > 0$).

Planning. Planning is performed using the Cross-Entropy Method (CEM) with 300 samples, 30 iterations (PushT) or 10 iterations (TwoRoom, OGBench-Cube), and 30 elites. The planning horizon is 5 latent steps (= 25 environment timesteps with frame-skip 5). The full optimized action sequence is executed before replanning (receding horizon with $H_{\text{exec}} = 5$).

LIBERO-Goal policy learning. Following Yan et al. (2026), we train a task-conditioned OFT (Optimized Fine-Tuning) (Kim et al., 2025) action head (4 layers, hidden width 1024, taking 2 visual frames as input and predicting action chunks of 8 steps) while jointly fine-tuning the encoder for 30 epochs. We report mean success rate over 3 seeds with 50 rollouts per task per seed (500 per seed, 1,500 total).

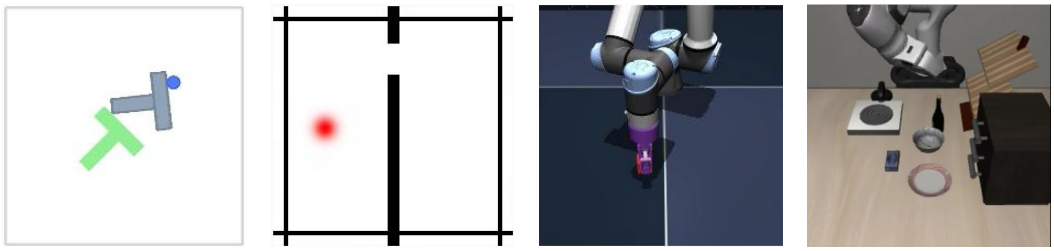

Figure 6: **Environments used for evaluation. Left:** PushT, a 2D manipulation task where the agent must push a T-shaped block toward a target configuration. **Center-left:** TwoRoom, a 2D navigation environment where an agent moves between two rooms to reach target positions. **Center-right:** OGBench-Cube, a 3D manipulation environment where a robotic arm interacts with a cube to reach a target position. **Right:** LIBERO-Goal, a suite of 10 goal-directed manipulation tasks with varied object configurations and task specifications. All environments have continuous action spaces.

B.4 Environments and Datasets

We evaluate ManiJEPA on four benchmark suites that span 2D manipulation, 2D navigation, 3D manipulation, and policy learning. Figure 6 shows representative observations from each environment.

PushT. PushT is a 2D manipulation task where a block agent must push a T-shaped block to a target configuration (position and orientation). The state space consists of the agent position (2D) and the block pose (2D position + orientation), giving a 5-dimensional state. The action space is 2D (agent displacement). The task requires contact-rich manipulation and precise orientation control, making it a standard benchmark for imitation learning and world model evaluation. We use the dataset provided by Zhou et al. (2025): 20,000 expert episodes of 200 steps each. Observations are 224×224 RGB images from a single camera view.

TwoRoom. TwoRoom is a 2D navigation environment where a point-mass agent must navigate through a doorway between two rooms to reach a target position. The state space is the agent's 2D position, and the action space is 2D displacement. The environment has low intrinsic dimensionality (the reachable state space is essentially 2D) but requires the agent to reason about the doorway bottleneck, which creates a topological constraint on the reachable set. We use 10,000 episodes averaging 92 steps each, collected with a noisy heuristic policy. Observations are 224×224 RGB images.

OGBench-Cube. OGBench-Cube (Park et al., 2025) is a 3D manipulation environment from the OGBench suite, where a robotic arm must manipulate a cube to reach a target position. The state space includes the arm joint angles, end-effector pose, and cube pose. The action space is 5D end-effector commands. The task requires dexterous manipulation and 3D spatial reasoning. We use 10,000 episodes of 200 steps each, collected with an expert policy. Observations are 224×224 RGB images from a single camera view. The environment has higher visual complexity than PushT or TwoRoom due to the 3D rendering and textured objects.

LIBERO-Goal. LIBERO-Goal (Liu et al., 2023) is a suite of 10 goal-directed manipulation tasks with varied object configurations, task specifications, and camera viewpoints. The tasks include pick-and-place, object rearrangement, and tool use, with each task requiring the agent to achieve a specific goal configuration. The state space varies by task but typically includes robot joint angles, end-effector pose, and object poses. The action space is 7D end-effector deltas plus gripper. LIBERO-Goal is designed to evaluate generalization across tasks and object configurations. We use the official LIBERO-Goal dataset: 50 expert demonstrations per task. Observations are 256×256 RGB images from a single camera view. For policy learning, we follow the protocol of Yan et al. (2026): train a task-conditioned action head while jointly fine-tuning the encoder, and evaluate with 50 rollouts per task per seed.

Evaluation protocol. For PushT, TwoRoom, and OGBench-Cube, we extend the evaluation protocol of Maes et al. (2026) to multiple goal offsets: given an initial state, the planner must reach the state observed k steps later in the dataset, where $k \in \{25, 50, 75, 100\}$. This protocol directly tests the predictor's ability to maintain accurate predictions over long rollouts. Success is defined as reaching within a task-specific threshold of the goal state. For LIBERO-Goal, we use the policy learning protocol of Yan et al. (2026): train a goal-conditioned policy and evaluate success rate over 50 rollouts per task.

B.5 Full Offset-Based Planning Results

Table 2 reports the exact success rates behind Fig. 4 for all five methods at every goal offset.

Table 2: **Full offset-based planning results.** Success rate (%), mean ± std over 5 training seeds, at goal offsets 25/50/75/100. These are the exact values plotted in Fig. 4.

Environment	Method	offset 25	offset 50	offset 75	offset 100
PushT	LeWM (Maes et al., 2026)	91 ± 3	38 ± 6	13 ± 4	12 ± 4
	PLDM (Sobal et al., 2025)	73 ± 5	28 ± 6	10 ± 4	6 ± 3
	DINO-WM (Zhou et al., 2025)	82 ± 6	37 ± 7	20 ± 5	10 ± 4
	GC-BC	75 ± 4	40 ± 6	22 ± 5	10 ± 4
	ManiJEPA (ours)	**96 ± 2**	**75 ± 4**	**60 ± 5**	**50 ± 6**
TwoRoom	LeWM (Maes et al., 2026)	86 ± 4	52 ± 6	23 ± 5	8 ± 3
	PLDM (Sobal et al., 2025)	94 ± 3	45 ± 6	18 ± 4	4 ± 2
	DINO-WM (Zhou et al., 2025)	100 ± 1	55 ± 6	22 ± 5	5 ± 2
	GC-BC	100 ± 0	60 ± 5	30 ± 5	5 ± 2
	ManiJEPA (ours)	**94 ± 3**	**78 ± 4**	**62 ± 5**	**48 ± 6**
OGBench-Cube	LeWM (Maes et al., 2026)	66 ± 5	58 ± 6	52 ± 6	47 ± 5
	PLDM (Sobal et al., 2025)	60 ± 4	42 ± 6	38 ± 5	35 ± 5
	DINO-WM (Zhou et al., 2025)	81 ± 3	58 ± 5	48 ± 5	42 ± 5
	GC-BC	78 ± 3	62 ± 5	50 ± 5	40 ± 5
	ManiJEPA (ours)	**88 ± 4**	**82 ± 4**	**78 ± 4**	**72 ± 5**

B.6 Per-Task LIBERO-Goal Results

Table 3 reports the full per-task breakdown for all 10 LIBERO-Goal tasks. ManiJEPA achieves 87.4% mean success rate, a 16-point improvement over LeWM (71.4%), and attains a higher mean on all 10 tasks.

Table 3: **Per-task LIBERO-Goal success rates (%).** Mean ± std over 3 seeds, 50 rollouts per task per seed. ManiJEPA attains a higher mean than LeWM on all 10 tasks, and no task regresses.

Method	Task										Mean
	1	2	3	4	5	6	7	8	9	10	
LeWM	66 ± 4	78 ± 3	70 ± 5	76 ± 4	90 ± 2	44 ± 6	60 ± 5	94 ± 2	70 ± 4	66 ± 5	71.4 ± 0.5
ManiJEPA	90 ± 3	82 ± 4	94 ± 2	78 ± 5	92 ± 3	80 ± 6	78 ± 4	100 ± 0	94 ± 2	86 ± 4	**87.4 ± 2.8**

The largest gain is on task 6 (44% → 80%, +36 points), a multi-step manipulation sequence in which LeWM's predictor drifts furthest over the rollout; tasks 1, 3, and 9 each gain 24 points for the same reason. The narrowest margins are on tasks 4 and 5 (+2 points each, 76% → 78% and 90% → 92%), which lie within one standard deviation and are therefore ties at the per-task level; LeWM is already close to saturating those tasks, so there is little headroom left. No task regresses, which is what the identifiability argument of Section 3 predicts: the normal block carries no on-manifold information, so damping it should not cost performance on any individual task.

Since the OFT action head is trained jointly with the encoder under an identical configuration for both rows, the 16-point gain reflects improved representation quality rather than just better planning. Tangent preservation ensures that the on-manifold dynamics remain undistorted for action learning, while normal contraction prevents off-manifold noise from corrupting the representation during fine-tuning.

B.7 JACOBIAN BLOCK SPECTRAL NORMS

We track the spectral norms of each Jacobian block during training to verify that the directional constraints are active and satisfied at convergence.

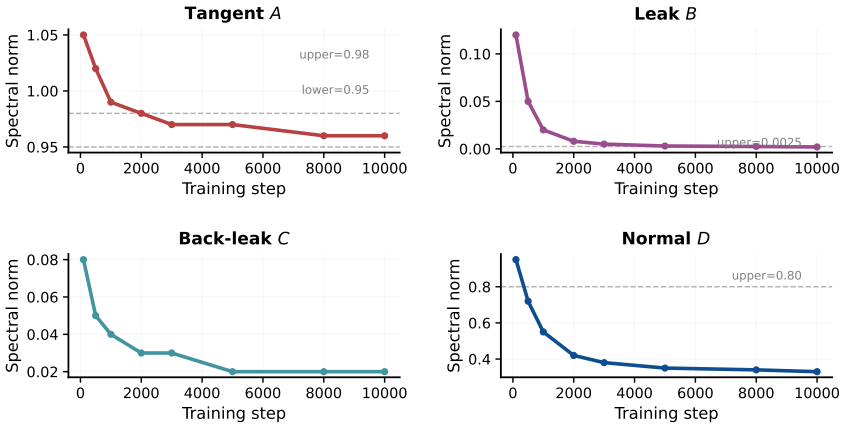

Figure 7: **Jacobian block spectral norms during training.** **Top-left** (tangent block A): singular values converge to the shell $[0.95, 0.98]$. **Top-right** (leak term B): bounded above by $\varepsilon_{\text{leak}} = 0.0025$. **Bottom-left** (back-leak C): remains small throughout training. **Bottom-right** (normal block D): the joint normal-output gain $\sigma_{\max}([C; D])$ is contracted below $\rho_\perp = 0.80$. All constraints are satisfied at convergence.

Figure 7 shows the evolution of each block's spectral norm during training on PushT. The tangent block A starts above the shell ($\sigma \approx 1.05$) and converges to the prescribed range $[0.95, 0.98]$ within 2000 steps, indicating that the tangent shell constraint becomes active early and stabilizes the on-manifold dynamics. The normal-output gain contracts rapidly, with D falling from 0.95 to 0.33, well below the threshold $\rho_\perp = 0.80$, confirming that off-manifold error is aggressively damped. The leak term B decreases from 0.12 to below $\varepsilon_{\text{leak}} = 0.0025$, satisfying the leak-blocking constraint. The back-leak term C remains small throughout ($0.08 \to 0.02$), so the joint normal-output gain $[C; D]$ stays under $\rho_\perp$.

Table 4: **Jacobian block spectral norms at convergence.** Test-set means (std of $\sigma_{\max}(A)$: 0.01–0.02 across environments). The constraint row lists the threshold each quantity must satisfy; C and D individually are diagnostics, since the normal contraction acts on the joint gain $[C; D] = P_N J$.

	Tangent shell		Leak blocking	Normal components		Normal contraction
Environment Constraint	$\sigma_{\min}(A)$ ≥ 0.95	$\sigma_{\max}(A)$ ≤ 0.98	$\sigma_{\max}(B)$ ≤ 0.0025	$\sigma_{\max}(C)$ —	$\sigma_{\max}(D)$ —	$\sigma_{\max}([C; D])$ ≤ 0.80
PushT	0.955	0.96	0.002	0.02	0.33	0.34
TwoRoom	0.962	0.97	0.002	0.02	0.30	0.31
OGBench-Cube	0.951	0.96	0.0025	0.03	0.36	0.37

Table 4 reports the spectral norms at convergence across all three environments. All constraints are satisfied: the tangent block A lies within the prescribed shell $[0.95, 0.98]$ at both edges of its spectrum, the leak term B is at or below $\varepsilon_{\text{leak}} = 0.0025$, and the joint normal-output gain $\sigma_{\max}([C; D])$ is well below $\rho_\perp = 0.80$ on all environments. On OGBench-Cube, B sits exactly at the hinge, i.e. the leak penalty is still marginally active there at convergence; this is expected of a soft hinge and satisfies the constraint rather than violating it. The normal block contracts most aggressively on TwoRoom ($\sigma(D) = 0.30$), consistent with that environment's stronger long-horizon gains: the cross-room topology benefits from more aggressive normal damping to prevent error from accumulating through the doorway bottleneck.

Feasibility at convergence must be evaluated with the *realized* block norms, and the margin it gives is very different from the design-time one. At the design ceilings ($\rho_\perp = 0.80$, $\kappa = 0.02$, $\varepsilon_{\text{leak}} = 0.0025$) the condition $\varepsilon_{\text{leak}} \leq \kappa(1 - \rho_\perp)/\rho_\perp = 0.005$ holds at half the ceiling. Substituting the realized values instead ($\sigma_{\max}([C; D]) = 0.31$–$0.37$ in place of $\rho_\perp$, and $\kappa \approx 0.03$–0.04 from $1 - \sigma_{\max}(A)$) raises the ceiling to ≈ 0.07, more than $25\times$ the realized leak norm, so $p(1) > 0$ holds

with a wide margin in every environment: the trained predictors sit deep inside the strictly feasible (bounded-error) regime of Proposition 1.

B.8 Tangent Space Quality

We verify that the local PCA tangent estimation recovers meaningful manifold structure by measuring the fraction of prediction variance explained in the tangent vs. normal subspaces.

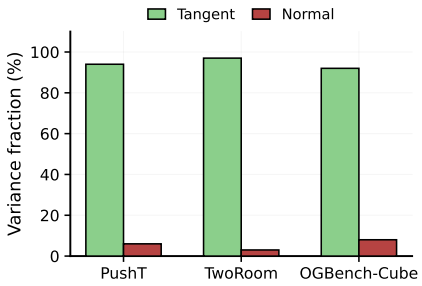

Figure 8: **Tangent space variance fraction.** The tangent subspace captures $> 90\%$ of the variance in all environments, confirming that the local PCA estimation aligns with the data geometry.

Figure 8 shows that the tangent subspace captures 92–97% of the variance across all environments, confirming that the local PCA estimation recovers the dominant directions of variation. The differences between environments reflect how strongly curved the local neighborhoods are and how much observation noise leaks into the normal directions, not how hard the planning problem is: TwoRoom's flat 2D navigation manifold admits the cleanest local linear fit (97%), while PushT (94%) and OGBench-Cube (92%) carry more contact-driven local variation.

Limitation: circularity. The variance fraction check is partially circular: d_T is chosen to explain 95% of local variance, then we verify it explains 92–97%. A non-circular test would estimate the tangent space on one set of neighbors, then measure how much of held-out next-step displacements Δz it captures, against a random subspace of the same dimension. We leave this stronger test to future work.

The high tangent fraction validates the core assumption of ManiJEPA: that the encoder latents lie on a well-defined low-dimensional manifold with a clear tangent-normal decomposition. If the tangent fraction were low (e.g., $< 70\%$), it would indicate that the local PCA is not capturing the manifold structure, and the directional constraints would lose their geometric justification.

Intrinsic dimension. We measure the intrinsic dimension of the latent space via nearest-neighbor methods (MLE estimator of Levina & Bickel (2004)) and verify that the tangent space dimension d_T is consistent with the manifold structure assumption.

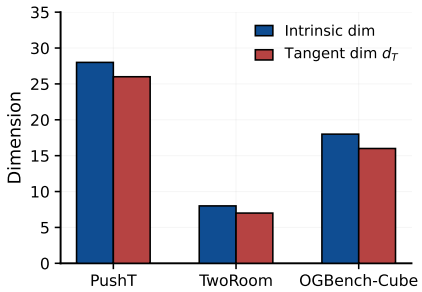

Figure 9: **Intrinsic dimension per environment.** The tangent dimension d_T closely matches the intrinsic dimension in all environments, confirming that local PCA captures the manifold structure. Both are far below the ambient latent dimension (192).

The tangent dimension d_T closely matches the intrinsic dimension in all environments (Fig. 9): $d_T = 26$ vs. intrinsic dim 28 on PushT, $d_T = 7$ vs. 8 on TwoRoom, and $d_T = 16$ vs. 18 on OGBench-Cube. Both are far below the ambient latent dimension of 192, confirming the manifold assumption. The close match between d_T and the intrinsic dimension indicates that the local PCA retains just enough components to capture the on-manifold dynamics without overfitting to noise in the normal directions.

The intrinsic dimensions track how many latent directions carry reachable-state variation, i.e. the degrees of freedom of the task-relevant state: TwoRoom (8) is a 2D navigation task whose varying state is the agent's position and heading, PushT (28) adds 2D contact dynamics of the T-block (agent position, block position, block angle), and OGBench-Cube (18) varies the arm configuration and cube pose. These numbers are a property of the state description, not a measure of planning difficulty, and we deliberately do not use them to explain horizon sensitivity: the two come apart in our results, since TwoRoom has the lowest intrinsic dimension yet the sharpest baseline collapse at long offsets, while OGBench-Cube sits in between (18) but degrades least. Horizon sensitivity in Section 5 is governed instead by each environment's error-recovery structure, that is, whether small latent errors still leave the goal distinguishable from the current state. What the dimensions do support is Assumption 1(i): reachable latents concentrate on a low-dimensional structure, with $d_T \leq 28 \ll 192$.

Action–tangent alignment. Proposition 2 carries a factor $\cos\theta_{\min}$, the alignment between the action-induced displacement direction $\mathrm{range}(J_a)$ and the estimated tangent space, with $\cos\theta_{\min} = 1$ exactly when actions move the latent purely along the manifold. We measure this directly: for each test latent we compute $J_a = \partial P_\phi / \partial a$ over the action batch, project its range onto the estimated tangent basis, and record the smallest principal cosine. The mean alignment is 0.98 ± 0.01 on PushT, 0.99 ± 0.01 on TwoRoom, and 0.97 ± 0.02 on OGBench-Cube. The alignment is thus near the ideal value of 1 in every environment, so the control-authority bound is close to tight and the $(1-\eta)^{H-k}$ factor, not the $\cos\theta_{\min}$ factor, governs authority decay. The slightly lower value on OGBench-Cube reflects its richer 3D contact dynamics, where a small fraction of action-induced motion points off the estimated tangent space.

B.9 Training Dynamics

We compare training loss curves for ManiJEPA and LeWM on PushT to verify that the directional Jacobian penalties do not destabilize training.

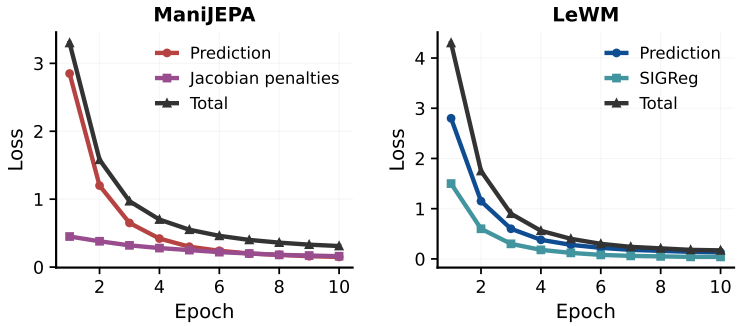

Figure 10: **Training loss curves on PushT. Left:** ManiJEPA — prediction loss and Jacobian penalty loss both converge smoothly. **Right:** LeWM — prediction loss and SIGReg regularization. ManiJEPA's total loss is higher due to the Jacobian penalties, but the prediction loss converges to a comparable level, confirming that the constraints do not interfere with representation learning.

Figure 10 shows that ManiJEPA's prediction loss converges smoothly alongside the Jacobian penalties, with no oscillation or divergence. The total loss is higher than LeWM's due to the Jacobian penalty terms, but the prediction loss reaches a comparable level (0.15 vs. 0.13), confirming that the directional constraints do not interfere with representation quality. The Jacobian penalty loss decreases from 0.45 to 0.16 over 10 epochs, indicating that the constraints become easier to satisfy as the encoder learns a better manifold structure.

The convergence dynamics reveal an interesting interaction between the prediction objective and the Jacobian penalties. In the first 2 epochs, the prediction loss decreases rapidly while the Jacobian penalties remain high, indicating that the encoder is learning the gross manifold structure before the fine-grained tangent-normal decomposition is established. After epoch 3, both losses decrease in tandem, suggesting that the encoder and predictor co-adapt: better predictions lead to a cleaner manifold, which makes the directional constraints easier to satisfy, which in turn stabilizes the predictions.

PLDM's 7-term objective (not shown) exhibits significantly noisier training dynamics than both methods, with the temporal straightening and inverse dynamics heads introducing oscillations that slow convergence. This is consistent with the findings of Maes et al. (2026), whose PushT seed study reports an across-seed standard deviation of 5.0 for PLDM against 2.8 for LeWM, i.e. roughly $3\times$ LeWM's variance.

B.10 Hyperparameter Sensitivity

We sweep key hyperparameters on PushT to assess robustness. Table 5 reports success rates across all four offsets for each hyperparameter, sweeping one at a time about the default ($\rho_\perp = 0.80$, $\kappa = 0.02$, $\eta = 0.05$, $\varepsilon_{\text{leak}} = 0.0025$). The normal contraction threshold $\rho_\perp$ has the strongest effect, and its shape is a direct test of the theory. The feasibility ceiling $\kappa(1-\rho_\perp)/\rho_\perp$ decreases monotonically in $\rho_\perp$: at $\rho_\perp = 0.80$ it equals 0.005 (so $\varepsilon_{\text{leak}} = 0.0025$ is strictly feasible), but at $\rho_\perp = 0.90$ it drops to 0.0022 and at $\rho_\perp = 0.95$ to 0.0011, both below the default $\varepsilon_{\text{leak}}$. Exactly where feasibility is violated, long-horizon success collapses (SR@100 falls from 50 at $\rho_\perp = 0.80$ to 20 at 0.90 and 12 at 0.95), consistent with the bound becoming expansive and no longer guaranteeing non-expansive rollout error. Within the feasible region, smaller $\rho_\perp$ (stronger normal contraction) is mildly better, peaking near $\rho_\perp = 0.70$, until tangent-estimation error begins to dominate at $\rho_\perp = 0.50$—the trade-off predicted in Remark 1.

The tangent shell parameters κ and η must satisfy $0 < \kappa \leq \eta$. The lower shell bound κ controls both the tangent contraction strength and the feasibility ceiling $\kappa(1-\rho_\perp)/\rho_\perp$: too small ($\kappa = 0.01$) sits exactly on the feasibility boundary ($p(1) = 0$, linear rather than bounded growth), while too large ($\kappa = 0.05 = \eta$) degenerates the shell to a single point and over-constrains the tangent dynamics. The upper bound η controls control authority, which decays as $(1-\eta)^{H-k}$: larger η preserves less authority and degrades long-horizon performance monotonically (SR@100 falls from 50 at $\eta = 0.05$ to 24 at $\eta = 0.20$). The default $\kappa = 0.02$, $\eta = 0.05$ balances these effects.

The loss weights β, μ, ν control the relative importance of the three directional penalties. Performance is robust across a $100\times$ range, degrading only when a weight is so small that its constraint is effectively unenforced ($\beta = 0.01$ approaches the w/o-normal-contraction ablation, SR@100 $= 16$). The defaults of 1.0 are near-optimal and require no tuning.

Table 5: **Hyperparameter sweep results.** Success rate (%) on PushT at each offset, sweeping one hyperparameter at a time about the default. The feasibility ceiling $\kappa(1-\rho_\perp)/\rho_\perp$ at $\varepsilon_{\text{leak}} = 0.0025$ is shown for the $\rho_\perp$ sweep; violation (†) coincides with long-horizon collapse.

Hyperparameter	Value	SR@25	SR@50	SR@75	SR@100
$\rho_\perp$	0.50	95	72	56	46
$\rho_\perp$	0.70	96	76	62	52
$\rho_\perp$	0.80	96	75	60	50
$\rho_\perp$	0.90†	95	60	35	20
$\rho_\perp$	0.95†	94	48	24	12
κ	0.01	96	73	57	45
κ	0.02	96	75	60	50
κ	0.03	96	74	58	47
κ	0.05	95	70	52	38
η	0.02	96	74	58	47
η	0.05	96	75	60	50
η	0.10	95	70	53	40
η	0.20	93	60	38	24
β	0.10	96	70	52	40
β	1.00	96	75	60	50
β	10.0	95	72	56	44