HAMILTONIAN JEPA: ACTION-CONDITIONED WORLD MODELS WITH AN INHERITED CONTROL STATE

Anonymous authors
Paper under double-blind review

ABSTRACT

Planning from pixels needs more than a latent space that is stable and predictable. The state the planner scores must also be organized by how actions move the system. Joint-embedding predictive architectures (JEPAs) avoid pixel reconstruction by predicting future representations, but existing action-conditioned JEPAs ask one embedding to serve both perception and control. We introduce H-JEPA, which separates the two. A wide perceptual code is regularized toward a well-scaled isotropic geometry with a Bures-Wasserstein prior, and a fixed orthonormal slice of that code is the control state, which inherits the code's covariance without any objective of its own. The state evolves under phase-conditioned dissipative port-Hamiltonian dynamics whose input port has orthonormal columns. Port-inverse consistency (PIC) reads the executed action back through the transpose of that port. We show that this readout is exactly the rollout error projected onto the port directions, so PIC is a parameter-free reweighting of prediction error and not an auxiliary action decoder. Untying the readout from the port breaks this identity and loses half of the gain. H-JEPA matches or exceeds reconstruction-free baselines, including the action-decoding Delta-JEPA, on four pixel-based control benchmarks after at most 10 training epochs, and its largest gain is on OGB-Cube (91.9 against 79.3 percent). Ablations on PushT and OGB-Cube separate the contributions of the structured predictor, PIC, the prediction horizon, the state rank, and the anti-collapse prior.

1 INTRODUCTION

An agent that can predict the consequences of its actions can plan: it scores candidate action sequences in imagination and executes the best one (Ha & Schmidhuber, 2018; Hafner et al., 2019). Trained on an offline corpus of interaction data, model-predictive control on a learned world model reaches goal images never demonstrated during training, with no reward function and no test-time gradient updates (Zhou et al., 2025; Sobal et al., 2025). The plan is only as good as the model's response to actions. A model that captures appearance but blurs the difference between action sequences ranks all plans alike, so action sensitivity is the central property of the learned dynamics.

Learning such models from pixels raises the question of what to predict. Reconstruction-based models regress future frames and can spend capacity on visual detail that is irrelevant to control (Watter et al., 2015; Hafner et al., 2019; 2020). Joint-embedding predictive architectures (JEPAs) instead predict future representations (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024), which removes the reconstruction target but exposes training to representation collapse. LeWM (Maes et al., 2026) addressed collapse for action-conditioned world models by attaching SIGReg (Balestriero & LeCun, 2025), a sketched Gaussian regularizer, to the embedding the planner scores.

This paper starts from the observation that stability is not the same as control. Matching a marginal embedding distribution certifies that no direction has died. It does not say which directions actions move, which coordinates should enter the planning metric, or how much capacity control requires. Three consequences follow. First, LeWM plans in the full embedding, so the metric that ranks candidates mixes control-relevant variation with dimensions that satisfy the prior without informing action selection. Second, the regularizer and the prediction loss act on the same coordinates, so their balance must be re-tuned whenever any other pressure changes. Third, latent-MSE objectives admit scale-degenerate solutions, since shrinking the targets makes prediction easy. Sub-JEPA (Zhao

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.

et al., 2026) found the full-space prior too restrictive for low-dimensional dynamics and relaxed it to random subspaces with clear gains, but its subspaces are views for the regularizer and not an extracted state.

H-JEPA (Hamiltonian JEPA) separates the perceptual code from the control state and lets the state inherit its geometry. A wide code $h \in \mathbb{R}^D$ is placed on a sphere by per-sample affine-free normalization and regularized toward isotropy with a Bures-Wasserstein moment penalty (Gelbrich, 1990). A fixed orthonormal projection U defines the state $s = U^\top h$, whose covariance is a fixed function of the code's covariance, so no separate state-space objective is needed. On this state we place a dissipative port-Hamiltonian step whose input port has orthonormal columns. *Port-inverse consistency* (PIC) reads the executed action back from the realized displacement through the transpose of the port, which is its exact pseudoinverse. The readout has no parameters and no free scale, and its residual equals the rollout error projected onto the port directions, so PIC reweights prediction error toward the directions in which actions act.

H-JEPA matches or exceeds reconstruction, full-space-prior, and action-decoding baselines on Two-Room, Reacher, PushT, and OGB-Cube under the same planning protocol, with the largest gain on OGB-Cube (91.9 percent against 79.3 for the strongest baseline). Our contributions are as follows. (i) A two-space world model in which one distributional objective shapes perception and a fixed orthonormal map carries covariance to the planner-facing state, with an exact description of the inherited covariance (§3.1 to §3.3). (ii) An analysis of anti-collapse penalties near vanishing variance, showing that the Bures-Wasserstein penalty keeps a constant restoring force on each standard deviation while SIGReg and the Euclidean covariance penalty lose theirs (§3.3). (iii) A phase-conditioned dissipative port-Hamiltonian predictor with an orthonormal input port, and PIC, which we show to be an exact error reweighting, with a control showing that the tie between readout and port accounts for half of its gain (§3.4, §3.5, §4.1). (iv) Planning, probing, and ablation results at a matched representation on two tasks, with diagnostics showing that the planner ranks candidates along directions beyond the probed physical variables (§4).

2 Related work

JEPA world models. LeCun (2022) proposed prediction in representation space as the basis of world models, validated by I-JEPA and V-JEPA (Assran et al., 2023; Bardes et al., 2024). LeJEPA (Balestriero & LeCun, 2025) motivates isotropic Gaussian embeddings and enforces them with SIGReg. LeWM (Maes et al., 2026) brought this program to action-conditioned world models from pixels, and Sub-JEPA (Zhao et al., 2026) improved planning by applying the Gaussian constraint in K random subspaces, with K selected per benchmark. Delta-JEPA (Zhang et al., 2026) adds action sensitivity through a learned linear decoder on latent differences, and sensorimotor world models (Ivashkov et al., 2026) make inverse dynamics the perception objective. H-JEPA differs in three respects. The subspace is a single fixed slice that *is* the control state, the prior stays full-space at LeWM's placement and reaches the state by inheritance, and the inverse readout is the forward model's own input port.

Latent dynamics models for control. E2C, DVBF, World Models, PlaNet, and Dreamer learn latent dynamics through pixel reconstruction and often reward prediction (Watter et al., 2015; Karl et al., 2017; Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020). DINO-WM plans with a frozen pretrained encoder (Zhou et al., 2025), and PLDM trains an end-to-end pixel JEPA with a VICReg-style objective (Sobal et al., 2025). H-JEPA uses neither pixels nor rewards as targets and trains its encoder end-to-end without teachers or stop-gradients (Grill et al., 2020). Its collapse defense is a moment-matching penalty in the Barlow Twins and VICReg family (Zbontar et al., 2021; Bardes et al., 2022), with the Euclidean covariance metric replaced by Bures-Wasserstein on Gaussians (Bhatia et al., 2019).

Structured dynamics. Hamiltonian neural networks learn a scalar energy whose gradient generates conservative flow (Greydanus et al., 2019), with controlled and generalized variants (Zhong et al., 2020; Course et al., 2020; Chen et al., 2021) and image-based generative forms (Toth et al., 2020). That literature targets a specified physical state or reconstruction. H-JEPA instead adopts the port-Hamiltonian form (van der Schaft & Jeltsema, 2014) as an inductive bias for a reconstruction-free representation trained for planning, and uses port orthonormality to remove a scale freedom that would otherwise interact with the representation objective.

Table 1: Design choices we relaxed during development, the failure each one produced in this architecture, and the constraint adopted instead. These observations motivate the constraints of §3 and complement the controlled ablations of §4.1.

Relaxed choice	Observed failure	Adopted constraint
BatchNorm in the projector, or a standardizer on s	Hides collapse: s keeps unit marginals while h degenerates to rank one	Per-sample affine-free LN_0, no normalizer on s (§3.1, §3.2)
LSUV-style whitening of the projector activations at initialization (Mishkin & Matas, 2016)	Amplifies projector weights about 10^4-fold, point-mass collapse within tens of steps	Isotropy as a loss with bounded per-step influence (§3.3)
Learnable projection U	Co-adapts with the losses and collapses the rank of the state	Fixed seeded orthonormal U (§3.2)
Learned action embedding	The PIC regression target collapses to effective rank about 1.3	Raw normalized actions (§3.1)
SIGReg, or the Euclidean penalty $\|\widehat{\Sigma} - I\|_F^2$	The restoring force fades as variance shrinks, collapse within about 50 to 100 steps	Bures-Wasserstein prior (§3.3)
Unconstrained port gain	The gain trades off against the encoder scale, so optimization moves scale and not content	Orthonormal port via differentiable QR (§3.4)

Inverse dynamics and state discovery. Recovering actions from transitions is a classical route to control-relevant features (Agrawal et al., 2016; Pathak et al., 2017), as are DeepMDP and bisimulation objectives (Gelada et al., 2019; Zhang et al., 2021). In exogenous-block MDPs, forward models with multi-step inverse prediction identify a control-endogenous state under stated assumptions (Lamb et al., 2023), with known limits on when multi-step inverse prediction is sufficient (Levine et al., 2024), and displacement-level action decoding appears in recent JEPA world models (Zhang et al., 2026; Ivashkov et al., 2026). PIC ties the readout to the transpose of the forward input map. Tying a readout to the transpose of an input map is standard practice for language-model embeddings (Press & Wolf, 2017), where the transpose is not the inverse of that map in general (Gu et al., 2026). In H-JEPA the port is constrained to be orthonormal, which makes the transpose its exact pseudoinverse. The bottleneck view relates to the information bottleneck (Tishby et al., 2000; Alemi et al., 2017), predictive state representations (Littman & Sutton, 2001), and balanced truncation (Moore, 1981).

3 METHOD

Setting. We are given offline trajectories of RGB frames x_t and continuous actions, with no rewards and no simulator state. The goal is a latent model in which a planner can roll out candidate action sequences and rank them by distance to an encoded goal image. An encoder produces a *perceptual code* h_t (§3.1). A fixed linear slice of the code is the *control state* s_t, the only quantity the planner scores (§3.2). One distributional prior acts on the code and keeps it from collapsing (§3.3). A port-Hamiltonian predictor advances the state under actions (§3.4). Training adds port-inverse consistency (PIC), which asks the predictor's own action-input map to recover the executed action from the observed displacement (§3.5). One model step is

$$h_t = E(x_t), \qquad s_t = U^\top h_t, \qquad v_t = s_t - s_{t-1}, \qquad c_t = [s_t, v_t], \qquad \hat{s}_{t+1} = \mathcal{F}_\theta(s_t, \bar{a}_t \mid c_t). \tag{1}$$

Reading equation 1 from left to right, E is the image encoder and $h_t \in \mathbb{R}^D$ is the code of the single frame x_t. The matrix $U \in \mathbb{R}^{D\times r}$ has orthonormal columns and is never trained, and $s_t \in \mathbb{R}^r$ with $r \le D$ is the control state. Since s_t is computed from one frame it carries no motion information, so we form the finite-difference velocity $v_t \in \mathbb{R}^r$ and concatenate it with the state into the *phase* $c_t \in \mathbb{R}^{2r}$ (square brackets denote concatenation). The action $\bar{a}_t \in \mathbb{R}^{d_a}$ is the normalized control applied between frames t and $t+1$. The predictor $\mathcal{F}_\theta$ with parameters θ advances s_t by one step under $\bar{a}_t$. The notation $\mathcal{F}_\theta(\cdot \mid c_t)$ means that the vector fields inside the predictor are functions of the phase, while only the state is advanced. The velocity is recomputed from consecutive states, with $v_0 = 0$ at the start of a window, and in open-loop rollouts it is formed from predictions, $\hat{v}_{t+i} = \hat{s}_{t+i} - \hat{s}_{t+i-1}$. An overview figure is given in Appendix A.

One rule organizes the design. The distributional objective acts only on perception, every map between the code and the planner is fixed or explicitly constrained, and no batch statistic or learned gain sits on that path. Each constraint answers a failure we observed when it was relaxed (Table 1).

3.1 Perception on a sphere

Following the LeWM perception stack, each frame is encoded independently as

$$h_t = \mathrm{LN}_0\big(\mathrm{MLP}(\mathrm{ViT}(x_t))\big) \in \mathbb{R}^D, \qquad \mathrm{LN}_0(z) = \sqrt{D}\,\frac{z - \mathrm{mean}(z)\mathbf{1}}{\|z - \mathrm{mean}(z)\mathbf{1}\|_2},$$

with a ViT-T/14 trained from scratch (Dosovitskiy et al., 2021) and an MLP projector at $D = 192$. LN_0 is LayerNorm without learned scale or bias. It centers each sample across its coordinates and fixes $\|h\|_2 = \sqrt{D}$, so every code satisfies $e^\top h = 0$ with $e = \mathbf{1}/\sqrt{D}$ and lies on a common sphere. Fixing the radius removes a scale freedom that would let the encoder inflate or shrink transition magnitudes instead of improving geometry, so the prior of §3.3 redistributes variance and cannot grow it.

Each model step spans m environment steps (the frameskip), and the action is the mean commanded control over that window, $\bar{a}_t = \frac{1}{m}\sum_{j=1}^{m} a_{t,j}$, in normalized coordinates. We use no learned action embedding, which co-adapted with the dynamics in our experiments (Table 1).

3.2 A control state with inherited covariance

The control state is $s_t = U^\top h_t \in \mathbb{R}^r$, where U is drawn once from a seeded QR factorization of a Gaussian matrix. Because U is fixed and linear,

$$\mathrm{Cov}(s) = U^\top \mathrm{Cov}(h)\, U, \tag{2}$$

so the covariance geometry imposed on the perceptual code is inherited by the control state without introducing a separate state-space objective or normalization. We refer to this property as *covariance inheritance*. At the prior's target $\mathrm{Cov}(h) = I - ee^\top$ (§3.3), Eq. equation 2 gives

$$\mathrm{Cov}(s) = I_r - (U^\top e)(U^\top e)^\top.$$

Thus the target covariance is isotropic up to a single exceptional direction, whose eigenvalue is $1 - |U^\top e|_2^2$. For a random orthonormal U, $|U^\top e|_2^2$ concentrates around r/D, so the deviation from isotropy is controlled directly by the relative state dimension. When $r < D$, the state is therefore nearly isotropic with one direction receiving the corresponding shrinkage. When $r = D$, the centering direction e is contained in the span of U, giving one exact null direction. Trained codes need not reach the prior exactly, and Appendix G reports the resulting covariance spectra and the predicted null direction. Keeping U fixed also prevents the projection from co-adapting with the dynamics, while experiments with a learnable U instead produced degenerate low-rank states (Table 1).

3.3 A Bures-Wasserstein isotropy prior

Let $\mathcal{H} = \{h_j\}_{j=1}^n$ be the codes of all frames in a batch, pooled over batch and time, with empirical mean $\hat{\mu} = \frac{1}{n}\sum_j h_j$ and covariance $\widehat{\Sigma} = \frac{1}{n}\sum_j (h_j - \hat{\mu})(h_j - \hat{\mu})^\top$. Since every code satisfies $e^\top h = 0$, the covariance always has a zero eigenvalue along e, and the most isotropic covariance available is the projector $P = I - ee^\top$ onto the hyperplane orthogonal to e. We penalize the squared 2-Wasserstein distance W_2^2 between the Gaussian with the empirical moments and the Gaussian with the target moments, which has the closed Bures-Wasserstein form (Gelbrich, 1990; Bhatia et al., 2019):

$$\boxed{\mathcal{L}_{\mathrm{iso}}(\mathcal{H}) = \frac{1}{D}\, W_2^2\big(\mathcal{N}(\hat{\mu}, \widehat{\Sigma}),\, \mathcal{N}(0, P)\big) = \frac{1}{D}\Big(\|\hat{\mu}\|_2^2 + \textstyle\sum_{i=1}^{D-1}\big(\sqrt{\hat{\lambda}_i} - 1\big)^2\Big)} \tag{3}$$

where $\hat{\lambda}_1, \ldots, \hat{\lambda}_{D-1} \geq 0$ are the eigenvalues of $\widehat{\Sigma}$ on that hyperplane. The second equality holds because $\widehat{\Sigma}$ and P commute. Equation 3 constrains only the first two moments and does not require h to be Gaussian. It is the model's only distributional objective.

Why Bures-Wasserstein. Write $\sigma_i = \sqrt{\lambda_i}$ for the standard deviation of the code along one covariance direction. In H-JEPA the $D-r$ code directions outside the slice receive no gradient from prediction or PIC, so the prior alone must hold them open, and what matters is the force a penalty exerts on σ_i as $\sigma_i \to 0$. The three candidates behave differently:

$$\underbrace{\tfrac{\partial}{\partial\sigma}(\sigma-1)^2 = 2(\sigma-1) \to -2}_{\text{Bures-Wasserstein}}, \quad \underbrace{\tfrac{\partial}{\partial\sigma}(\sigma^2-1)^2 = 4\sigma(\sigma^2-1) \to 0}_{\text{Euclidean } \|\widehat{\Sigma}-I\|_F^2}, \quad \underbrace{\tfrac{\partial}{\partial\sigma}T(\sigma) = O(\sigma) \to 0}_{\text{SIGReg}}. \tag{4}$$

Bures-Wasserstein is a quadratic penalty on standard deviations, close in spirit to the variance term of VICReg (Bardes et al., 2022), and keeps a constant pull on a dying direction. The Euclidean penalty is quadratic in the variance, so its pull fades linearly with σ. SIGReg, the sketched Epps-Pulley normality statistic T (EPPS & PULLEY, 1983) used by LeJEPA and LeWM, is a smooth bounded function of the samples and is therefore also quadratic in σ near a point mass (Appendix H). This matches what we saw in training. In collapse runs within H-JEPA the SIGReg score saturated as the code concentrated and its gradient became negligible, exactly where latent prediction admits a constant solution, and Table 4 shows the same contrast in planning success.

3.4 Phase-conditioned port-Hamiltonian dynamics

Instead of an unconstrained transition map, the predictor has the form of a port-Hamiltonian system (van der Schaft & Jeltsema, 2014): autonomous motion is generated by a scalar energy and damped by a dissipation term, and the action enters through a separate input map called the *port*. We use this form as an inductive bias and do not assume that the latent coordinates are physical positions and momenta. For even r, the predictor consists of

$$J = \begin{bmatrix} 0 & -I_{r/2} \\ I_{r/2} & 0 \end{bmatrix}, \qquad H_\theta : \mathbb{R}^{2r} \to \mathbb{R}, \qquad R_\theta(c) = L_\theta(c)L_\theta(c)^\top \succeq 0, \qquad G_\theta(c) \in \mathbb{R}^{r\times d_a}. \tag{5}$$

Here $J \in \mathbb{R}^{r\times r}$ is the constant canonical symplectic matrix, with $I_{r/2}$ the identity of size $r/2$. It is skew-symmetric, so the flow $J\nabla_s H$ conserves H in continuous time. H_θ is a scalar energy computed by an MLP from the phase $c = [s, v]$, and $\nabla_s H_\theta$ denotes its gradient with respect to the state argument only, with v treated as a constant input. $L_\theta(c) \in \mathbb{R}^{r\times r}$ is a lower-triangular matrix output by a second MLP, so the dissipation matrix $R_\theta(c)$ is positive semidefinite by construction and can only remove energy. $G_\theta(c)$ is the input port, which maps an action in $\mathbb{R}^{d_a}$ to a state displacement. It is the output of a third MLP followed by a differentiable thin QR factorization, so its columns are orthonormal, $G_\theta(c)^\top G_\theta(c) = I_{d_a}$ (it lies on the Stiefel manifold). All three networks take the phase as input, which is how velocity information reaches the dynamics. The continuous-time field $\dot{s} = (J - R_\theta(c))\nabla_s H_\theta(s, v) + G_\theta(c)\bar{a}$ is discretized with one explicit Euler step of size $\Delta t = 1$:

$$\hat{s}_{t+1} = \mathcal{F}_\theta\big(s_t, \bar{a}_t \mid c_t\big) = s_t + \Delta t\Big[\underbrace{\big(J - R_\theta(c_t)\big)\nabla_s H_\theta(s_t, v_t)}_{\text{passive flow}} + \underbrace{G_\theta(c_t)\,\bar{a}_t}_{\text{control flow}}\Big]. \tag{6}$$

Continuous-time energy properties need not hold exactly after Euler discretization (§5).

The orthonormal port fixes action geometry and scale. An unconstrained input map $B(c)$ can be written as $G(c)S(c)$ with orthonormal G and gain S. The gain can trade off against latent scale and PIC readout, causing scale drift without improving content (Table 1). Constraining the port to orthonormal columns removes this freedom. The action Jacobian $\partial\mathcal{F}/\partial\bar{a} = \Delta t\, G_\theta(c)$ then has all singular values equal to Δt, so one unit of normalized action produces one unit of direct latent displacement. State-dependent control effectiveness is therefore expressed through the port orientation and state geometry (Murray et al., 1994).

3.5 Training objective

Training uses three terms,

$$\boxed{\mathcal{L} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{PIC}} + \mathcal{L}_{\text{iso}}(\mathcal{H})}, \tag{7}$$

with no pixel reconstruction, learned inverse model, contrastive loss, teacher network, or normalization of s.

Open-loop prediction. We supervise the same multi-step composition the planner uses. Let $\mathcal{T}$ be the set of rollout start times in a batch. From an encoded state s_t, the model rolls forward for K_p steps under the executed actions, feeding back its own predictions, and matches the encoded future at every step:

$$\mathcal{L}_{\text{pred}} = \frac{1}{|\mathcal{T}|K_p} \sum_{t\in\mathcal{T}} \sum_{i=1}^{K_p} \tfrac{1}{r} \left\|\delta_{t+i}\right\|_2^2, \quad \delta_{t+i} = s_{t+i} - \hat{s}_{t+i}, \quad \hat{s}_{t+i} = \mathcal{F}_\theta\big(\hat{s}_{t+i-1}, \bar{a}_{t+i-1} \mid \hat{c}_{t+i-1}\big), \tag{8}$$

with $\hat{s}_t = s_t$ and $\hat{c}_{t+i} = [\hat{s}_{t+i}, \hat{v}_{t+i}]$. Squared errors are averaged over coordinates, so all three terms are of order one. For $K_p = 1$ this is the next-latent regression of LeWM. We use $K_p = 5$, equal to the planning horizon, so training penalizes the composed rollouts that the planner ranks.

Port-inverse consistency (PIC). The port maps an action into a state displacement. PIC asks the same port, transposed, to map the observed displacement back to the action. At rollout step i, we subtract the passive flow from the observed displacement and read the remainder through the transpose of the port:

$$\hat{u}_{t+i} = G_\theta(\hat{c}_{t+i})^\top \left[\frac{s_{t+i+1} - \hat{s}_{t+i}}{\Delta t} - \big(J - R_\theta(\hat{c}_{t+i})\big) \nabla_s H_\theta\big(\hat{s}_{t+i}, \hat{v}_{t+i}\big) \right], \tag{9}$$

and penalize its distance to the executed action over K_{PIC} rollout steps:

$$\boxed{\mathcal{L}_{\text{PIC}} = \frac{1}{|\mathcal{T}|K_{\text{PIC}}} \sum_{t\in\mathcal{T}} \sum_{i=0}^{K_{\text{PIC}}-1} \tfrac{1}{d_a} \left\|\hat{u}_{t+i} - \bar{a}_{t+i}\right\|_2^2}. \tag{10}$$

Because the port has orthonormal columns, $G_\theta^\top$ is its exact pseudoinverse, as in tied input-output maps (Press & Wolf, 2017), so the readout needs no learned inverse model and has no scale of its own. Substituting the Euler step equation 6 into equation 9 (Appendix B) gives

$$\hat{u}_{t+i} - \bar{a}_{t+i} = \tfrac{1}{\Delta t}\, G_\theta(\hat{c}_{t+i})^\top \delta_{t+i+1}, \tag{11}$$

where δ_{t+i+1} is the rollout error of equation 8. The prediction and PIC terms therefore combine into a single quadratic form in that error,

$$\mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{PIC}} = \frac{1}{|\mathcal{T}|K_p\, r} \sum_{t\in\mathcal{T}} \sum_{i=1}^{K_p} \delta_{t+i}^\top \Big(I + \tfrac{r}{d_a\, \Delta t^2}\, G_{t+i-1} G_{t+i-1}^\top \Big) \delta_{t+i}, \tag{12}$$

when $K_{\text{PIC}} = K_p$, with $G_{t+i-1} = G_\theta(\hat{c}_{t+i-1})$.

Equation 12 tells us what PIC actually measures. PIC is not a separate task of predicting actions. It is the same rollout error as $\mathcal{L}_{\text{pred}}$, restricted to the d_a port directions through which the action enters the dynamics. Adding the two losses therefore gives a single rollout error, measured with a state-dependent metric that counts errors along the port directions $1 + r/(d_a \Delta t^2)$ times more than errors along any other direction.

This view also explains why the model cannot satisfy PIC through a shortcut. There are two obvious shortcuts, and neither works. The first is to rescale the readout until actions become easy to decode. This fails because actions are in fixed normalized units, the port has unit singular values, and the scale of s is inherited rather than set by the readout. The second is to rotate the port away from the directions where the rollout is wrong. This fails because the same port also injects the action in Equation equation 6, so a port that no longer points at the error makes the prediction worse and raises $\mathcal{L}_{\text{pred}}$.

Both arguments rest on one design choice: the action is read out through the same port G_θ that injected it. §4.1 tests this directly by replacing $G_\theta^\top$ in the readout with a fixed random orthonormal matrix, which breaks the link between injection and readout. We use $K_{\text{PIC}} = K_p = 5$ throughout.

Planning. At test time we plan with the cross-entropy method (CEM) in a model-predictive control loop (Rubinstein, 1999). The goal image is encoded to $s_{\text{goal}} = U^\top E(x_{\text{goal}})$, candidate action sequences are rolled out with equation 6 for K_p steps, and each is scored by the terminal distance $\|\hat{s}_{t+K_p} - s_{\text{goal}}\|_2^2$. The phase conditions the dynamics but is not part of the cost. Main results follow the Delta-JEPA budget of 500 episodes. Controlled ablations use the lower-cost LeWM budget with otherwise identical planning settings.

Table 2: Planning success rate (%) on the four LeWM benchmarks over 500 test episodes per task, mean ± std over three training seeds. Baselines are evaluated under the Delta-JEPA protocol, in which Delta-JEPA trains for 50 epochs. H-JEPA trains for 10 epochs and is evaluated at the per-task checkpoint epoch selected on a held-out seed (§4).

Method	Two-Room	Reacher	PushT	OGB-Cube
PLDM	93.73 ± 1.03	64.33 ± 2.14	76.13 ± 1.70	57.27 ± 1.53
LeWM	74.93 ± 0.42	79.87 ± 0.90	84.53 ± 1.50	64.13 ± 1.89
Sub-JEPA	90.60 ± 0.53	81.00 ± 2.40	63.73 ± 0.12	62.67 ± 1.45
Delta-JEPA	$\mathbf{100.00 \pm 0.00}$	81.33 ± 0.50	89.07 ± 1.90	79.27 ± 1.81
H-JEPA (ours)	$\mathbf{100.00 \pm 0.00}$	$\mathbf{86.13 \pm 0.23}$	$\mathbf{90.40 \pm 0.60}$	$\mathbf{91.93 \pm 1.30}$

4 EXPERIMENTS

Setup. We evaluate on the four pixel-based planning benchmarks of the LeWM suite: Two-Room navigation (Maes et al., 2026), DeepMind Control Reacher (Tassa et al., 2018), PushT (Chi et al., 2025), and OGBench Cube manipulation (Park et al., 2025), hereafter OGB-Cube. All models train on the same offline image-action trajectories. We compare against PLDM (Sobal et al., 2025), LeWM (Maes et al., 2026), Sub-JEPA (Zhao et al., 2026), and Delta-JEPA (Zhang et al., 2026) under the Delta-JEPA evaluation protocol of 500 test episodes per task (Zhang et al., 2026), in which Delta-JEPA is trained from scratch for 50 epochs. We train H-JEPA for 10 epochs on every task. One held-out pilot seed, excluded from the reported averages, is used only to choose which epoch's checkpoint to evaluate on each task, and this choice is then fixed for the remaining seeds: epoch 8 on Two-Room, 6 on Reacher, 10 on PushT, and 3 on OGB-Cube. H-JEPA uses AdamW (Loshchilov & Hutter, 2019), windows of $T = 2 + K_p = 7$ frames, and $K_p = K_{\mathrm{PIC}} = 5$. The only task-specific architectural choice is the state rank: 32 on Two-Room and OGB-Cube, 64 on Reacher, and 192 on PushT, chosen once from the lower-budget sweep on a held-out seed in §4.1 and then fixed for the 500-episode evaluation. Per-benchmark capacity selection is common in this suite: Sub-JEPA selects its number of subspaces per task ($K = 32$, except $K = 16$ on PushT, where $K = 32$ fails) (Zhao et al., 2026). All other hyperparameters are shared (Appendix C).

Planning results. Table 2 reports planning success. H-JEPA is best or tied on every task. It saturates Two-Room together with Delta-JEPA, is within one standard deviation of Delta-JEPA on PushT, and leads by 4.8 points on Reacher and 12.7 points on OGB-Cube. H-JEPA trains for 10 epochs, against 50 for Delta-JEPA. The Sub-JEPA paper reports higher numbers for itself (95.0, 84.0, 89.0, and 76.3 on the four tasks) under a different protocol with 50 evaluation episodes, against 500 here (Zhao et al., 2026). H-JEPA is above those numbers as well. As a check on identical episodes, a paired comparison against the official LeWM checkpoint on N=200 matched episodes gives 88.5 against 82.0 on PushT (McNemar $p \approx 0.053$). Absolute rates in this paired test are not comparable to Table 2. Because Table 2 changes representation and predictor together, §4.1 separates them.

What the state encodes. Physical probing (Appendix D) shows a consistent division of labor. H-JEPA is strongest on the directly controlled variable (PushT agent location, linear MSE 0.028 against 0.062 and 0.065), while LeWM and Sub-JEPA decode the PushT block pose better. On OGB-Cube, H-JEPA is far stronger on joint velocity and end-effector position, and the methods are at parity on Reacher. Planning quality is thus not monotone in object-pose decodability.

What the predictor learns. Appendix E audits the learned fields on held-out trajectories. The balance between passive and control flow in equation 6 is task dependent. On PushT the control flow is only 0.13 times the passive flow, the passive flow is dominated by dissipation, and linear probes from the passive flow, control flow, and energy gradient each explain 88 to 90 percent of the physical state increment. On OGB-Cube the control flow is 1.53 times the passive flow. On Reacher and OGB-Cube the component-wise physical alignment is weaker, even though task variables remain strongly encoded in s.

Table 3: Predictor class on PushT at a matched representation and horizon (K_p=5 in all rows). Parameters are total model followed by dynamics module. H-JEPA rows share the ViT-T/14 encoder and 0.79M projector.

Predictor	Params total / dyn.	Success	Representation
LeWM, native embedding	18.0M / 10.8M	20%	SIGReg
AdaLN transformer on H-JEPA, r=64	9.6M / 3.3M	84%	H-JEPA
Port-Hamiltonian + PIC, r=64	7.2M / 0.86M	92%	H-JEPA
Port-Hamiltonian + PIC, r=192	11.6M / 5.4M	**94**%	H-JEPA

Table 4: Mechanism ablations, success (%) at N=300. H is the port-Hamiltonian predictor. †Value from Table 2.

Configuration	PushT	Cube
H + PIC, prior on h (full)	**93.3**	**91.9**†
MLP + PIC	85.0	84.3
H, no PIC	75.0	89.0
MLP, no PIC	32.0	
H + PIC, untied readout Q	83.7	
H + PIC, prior on s	86.3	
H + PIC, SIGReg prior	36.7	

Table 5: Horizons on PushT, success (%).

(K_p, K_{PIC})	$(1,0)$	$(1,1)$	$(3,3)$	$(5,1)$	$(5,5)$
Success	12	34	**94**	90	**94**

Table 6: State rank sweep, success (%).

Task	r=16	r=32	r=64	r=192
Two-Room		**100**	100	100
Reacher	72 (80)	82 (84)	**94**	78
PushT		84 (81)	92	**94**
OGB-Cube		**96**		76, 78

4.1 Ablations and analysis

Predictor class at a matched representation. The JEPA baselines use a causal AdaLN transformer over three latents. We keep the H-JEPA perception stack, the fixed U, the state s, and K_p=5, and swap only the dynamics for that transformer (Table 3). With K_p=5, LeWM's transformer on its native SIGReg embedding reaches 20 percent. The same transformer on the H-JEPA representation reaches 84 percent. Replacing it by the port-Hamiltonian predictor with PIC raises success to 92 percent at r=64 while shrinking the dynamics module from 3.3M to 0.86M parameters. The selected r=192 model reaches 94 percent with 11.6M parameters in total, below the 18.0M of LeWM. The last swap changes temporal-state construction and action injection together with the predictor, so it compares complete predictor formulations at a matched representation and horizon.

Structured dynamics and PIC. Table 4 isolates the remaining choices at matched rank on PushT (r=64, epoch 10) and OGB-Cube (r=32, seed 11), both at N=300. The structured predictor helps by a similar amount on both: with PIC in place, replacing the port-Hamiltonian step by an unstructured MLP costs 8.3 points on PushT (93.3 to 85.0) and 7.6 on OGB-Cube (91.9 to 84.3). The contribution of PIC is task dependent. Removing it costs 18.3 points on PushT and 2.9 on OGB-Cube, where the full-model reference is the main-benchmark value and the gap is within evaluation noise at N=300. This agrees with the reading of PIC as a reweighting equation 12 and with the predictor audit. On PushT the control flow is a small part of each displacement (0.13 of the passive flow), so an unweighted rollout loss is dominated by passive error and says little about the action-coupled part, and extra weight on the port directions matters. On OGB-Cube the control flow is the larger part (1.53), so the rollout loss already emphasizes it, and the extra weight from PIC is also smaller there (7.4 against 33, since $r = 32$ and $d_a = 5$). Without the Hamiltonian structure, PIC matters most: the MLP falls from 85.0 to 32.0 on PushT without it.

Tying the readout to the port. To test the tie between readout and port, we keep the full recipe, including the action target and the learned port G_θ in the forward step, and replace only $G_\theta^\top$ in the readout equation 9 by $Q^\top$ for a fixed random orthonormal $Q \in \mathbb{R}^{r \times d_a}$. The identity equation 11 then fails. The residual gains a term that vanishes only when the port coincides with Q (Appendix B), which competes with the state-dependent orientation that prediction asks of the port. This untied readout reaches 83.7 percent, against 75.0 without PIC and 93.3 with the tied readout. An action readout through a fixed orthonormal map therefore already helps, and tying it to the port, which turns it into a pure reweighting of rollout error, accounts for the other half of the gain.

Prior type and placement. Moving the Bures-Wasserstein prior from the code h onto the scored state s lowers PushT success from 93.3 to 86.3, which supports regularizing perception and letting the state inherit. Replacing Bures-Wasserstein by SIGReg on h reaches 36.7 percent at the same epoch 10, consistent with equation 4. This is a statement about SIGReg inside H-JEPA, where the prior alone must hold open the code directions outside the slice. It is not a statement about LeWM, where SIGReg acts on the same coordinates as the prediction loss.

Training at the planning horizon. Table 5 (PushT, r=192, N=50, epoch 10) shows that one-step prediction does not compose into a plannable rollout, with or without one-step PIC (12 and 34 percent). Matching both horizons at three or five steps reaches 94 percent. Cutting PIC to one step while keeping K_p=5 gives 90 percent. We therefore set $K_p = K_{\mathrm{PIC}}$ to the planning horizon.

Rank is the capacity knob. Table 6 shows that the useful control dimension is task dependent (bold: rank used in Table 2, parentheses: larger evaluation). Two-Room saturates at every tested rank, Reacher and OGB-Cube degrade at full rank, and PushT improves up to r=D. At $r = D$ the slice compresses nothing but still separates the scored coordinates from the regularized ones. The selected ranks are capacity choices for the planning metric and not estimates of physical state dimension. An audit of the CEM cost on the r=64 PushT checkpoint makes this concrete. On the candidate sets scored during evaluation, we fit a linear map from the probed physical variables of Appendix D to the terminal state, split every terminal state into that reconstruction and the orthogonal remainder, and rank candidates by the terminal cost computed from each part alone. The remainder alone has Spearman correlation 0.997 with the full candidate ranking and recovers 89 percent of the CEM elites, against 0.66 and 38 percent for the physical part. Planning therefore uses predictive directions beyond the variables exposed by standard physical probes.

Appendix F reports the history ablation that led to the parameter-free phase, and Appendix I the training curves.

5 Limitations

H-JEPA uses mechanics as a structural bias and makes no claim of latent system identification. The learned coordinates are free to rotate around the planning objective, and the audit in Appendix E confirms that component-level physical alignment is task dependent. The explicit Euler step does not preserve the energy properties of the continuous-time field, which our short-horizon planner did not require. The orthonormal port fixes the singular values of the instantaneous action map. This removes a scale ambiguity and makes PIC well conditioned, but it gives up state-dependent action gain. The state rank is selected per task and not learned, and at $r = D$ the state has one null direction (§3.2). The inherited covariance is only as isotropic as the trained code, and the measured spectra in Appendix G remain spread around one. Finally, our evidence is limited to offline pixel control in simulation with short-horizon CEM planning. Longer horizons, real observations, and online data collection remain to be tested.

6 Conclusion

We introduced H-JEPA, a world model that learns to plan from pixels without reconstruction by separating perceptual representation learning from the state used for control. A Bures-Wasserstein prior regularizes the perceptual code toward isotropy, while a fixed orthonormal projection transfers its covariance geometry to a control state with task-dependent dimension. A phase-conditioned port-Hamiltonian predictor structures the latent dynamics, and tied port-inverse consistency yields an exact reweighting of rollout error along action-input directions without an additional decoder. Ablations support the benefits of structured dynamics and, on PushT, regularizing perception and tying the inverse readout to the action port. H-JEPA matches or exceeds all evaluated baselines across four pixel-based control benchmarks. Together, these results establish covariance inheritance and structured, action-sensitive prediction as a promising combination for learning representations that support effective planning.

AI USE STATEMENT

Large language models were used for limited language editing, grammar refinement, minor wording and presentation suggestions, and code auto-completion during implementation. Any such suggestions were independently reviewed, validated where applicable, and edited by the authors before inclusion. All research ideas, methodology, derivations, experiments, analyses, and conclusions were developed and verified by the authors, who take full responsibility for the content of the paper.

ETHICS STATEMENT

This work studies representation learning and planning in standard simulated control benchmarks. It involves no human subjects, no personal data, and no dataset releases beyond configurations for public benchmarks. Although the present experiments are confined to simulation, deploying learned world models on physical systems would require application-specific validation of safety, robustness, and failure behavior.

REPRODUCIBILITY STATEMENT

Section 3 specifies the model, training objective, and planning procedure, with equation 3, equation 6, equation 8, and equation 10 defining the distributional objective, dynamics update, multi-step prediction loss, and PIC loss. Appendix B derives the PIC identity. Default hyperparameters and the per-task checkpoint epochs are listed in Appendix C. The probing protocol follows Zhang et al. (2026) and Maes et al. (2026) and is restated in Appendix D. Predictor diagnostics are in Appendix E, the geometry and collapse analyses in Appendices G and H, and optimization traces in Appendix I. An anonymized code release with training configurations for all reported runs is included as supplementary material.

REFERENCES

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/c203d8a151612acf12457e4d67635a95-Paper.pdf.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HyxQzBceg.

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, Los Alamitos, CA, USA, June 2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.01499. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.01499.

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

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2. Featured Certification.

Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the bures–wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2):165–191, 2019. ISSN 0723-0869.

doi: https://doi.org/10.1016/j.exmath.2018.01.002. URL https://www.sciencedirect.com/science/article/pii/S0723086918300021.

Yuhan Chen, Takashi Matsubara, and Takaharu Yaguchi. Neural symplectic form: Learning hamiltonian equations on general coordinate systems. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 16659–16670. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/8b519f198dd26772e3e82874826b04aa-Paper.pdf.

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. doi: 10.1177/02783649241273668. URL https://doi.org/10.1177/02783649241273668.

Kevin Course, Trefor Evans, and Prasanth Nair. Weak form generalized hamiltonian learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18716–18726. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d93c96e6a23fff65b91b900aaa541998-Paper.pdf.

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. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

T. W. EPPS and LAWRENCE B. PULLEY. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 12 1983. ISSN 0006-3444. doi: 10.1093/biomet/70.3.723. URL https://doi.org/10.1093/biomet/70.3.723.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G. Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2170–2179. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/gelada19a.html.

Matthias Gelbrich. On a formula for the l2 wasserstein metric between measures on euclidean and hilbert spaces. *Mathematische Nachrichten*, 147(1):185–203, 1990. doi: https://doi.org/10.1002/mana.19901470121. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mana.19901470121.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/26cd8ecadce0d4efd6cc8a8725cbd1f8-Paper.pdf.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

Jian Gu, Aldeida Aleti, Chunyang Chen, and Hongyu Zhang. Rethinking weight tying: Pseudo-inverse tying for lm stable training and updates, 2026. URL https://arxiv.org/abs/2602.04556.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.),

Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

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.

Petr Ivashkov, Randall Balestriero, and Bernhard Schölkopf. Sensorimotor world models: Perception for action via inverse dynamics, 2026. URL https://arxiv.org/abs/2606.20104.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HyTqHL5xg.

Alex Lamb, Riashat Islam, Yonathan Efroni, Aniket Rajiv Didolkar, Dipendra Misra, Dylan J Foster, Lekan P Molu, Rajan Chari, Akshay Krishnamurthy, and John Langford. Guaranteed discovery of control-endogenous latent states with multi-step inverse models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=TNocbXm5MZ.

Yann LeCun. A path towards autonomous machine intelligence. *OpenReview preprint*, 2022.

Alexander Levine, Peter Stone, and Amy Zhang. Multistep inverse is not all you need. In *Reinforcement Learning Conference*, 2024. URL https://openreview.net/forum?id=xyrgG4rsqY.

Michael Littman and Richard S Sutton. Predictive representations of state. In T. Dietterich, S. Becker, and Z. Ghahramani (eds.), *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/1e4d36177d71bbb3558e43af9577d70e-Paper.pdf.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

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.

V A Marčenko and L A Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, apr 1967. doi: 10.1070/SM1967v001n04ABEH001994. URL https://doi.org/10.1070/SM1967v001n04ABEH001994.

Dmytro Mishkin and Jiri Matas. All you need is a good init, 2016. URL https://arxiv.org/abs/1511.06422.

B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981. doi: 10.1109/TAC.1981.1102568.

Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 94937–94982, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ecd92623ac899357312aaa8915853699-Paper-Conference.pdf.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/pathak17a.html.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In Mirella Lapata, Phil Blunsom, and Alexander Koller (eds.), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2025/.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, September 1999. doi: 10.1023/A:1010091220143. URL https://ideas.repec.org/a/spr/metcap/v1y1999i2d10.1023_a1010091220143.html.

Uladzislau 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. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 43905–43941. Curran Associates, Inc., 2025. doi: 10.52202/085713-1465. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/3e7cf447f21cd11c846463affefce665-Paper-Conference.pdf.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018. URL https://arxiv.org/abs/1801.00690.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000. URL https://arxiv.org/abs/physics/0004057.

Peter Toth, Danilo J. Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJenn6VFvB.

Arjan van der Schaft and Dimitri Jeltsema. Port-hamiltonian systems theory: An introductory overview. *Foundations and Trends in Systems and Control*, 1(2-3):173–378, 06 2014. ISSN 2325-6818. doi: 10.1561/2600000002. URL https://doi.org/10.1561/2600000002.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/a1afc58c6ca9540d057299ec3016d726-Paper.pdf.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12310–12320. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zbontar21a.html.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

Zhenghao Zhang, Yuanxiang Wang, Zhenyu Guan, Yujia Yang, Bingkang Shi, Tianyu Zong, Hongzhu Yi, Guoqing Chao, Xingchen Chen, Tiankun Yang, Chenxi Bao, Tao Yu, Jingjing Zhou, and Jungang Xu. Delta-jepa: Learning action-sensitive world models via latent difference decoding, 2026. URL https://arxiv.org/abs/2606.31232.

Kai Zhao, Dongliang Nie, Yuchen Lin, Zhehan Luo, Yixiao Gu, Deng-Ping Fan, and Dan Zeng. Sub-jepa: Subspace gaussian regularization for stable end-to-end world models, 2026. URL https://arxiv.org/abs/2605.09241.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxmb1rKDS.

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 Architecture overview

Figure 1 summarizes the training-time model: perception on a sphere, the fixed orthonormal slice, the port-Hamiltonian predictor, and the three losses.

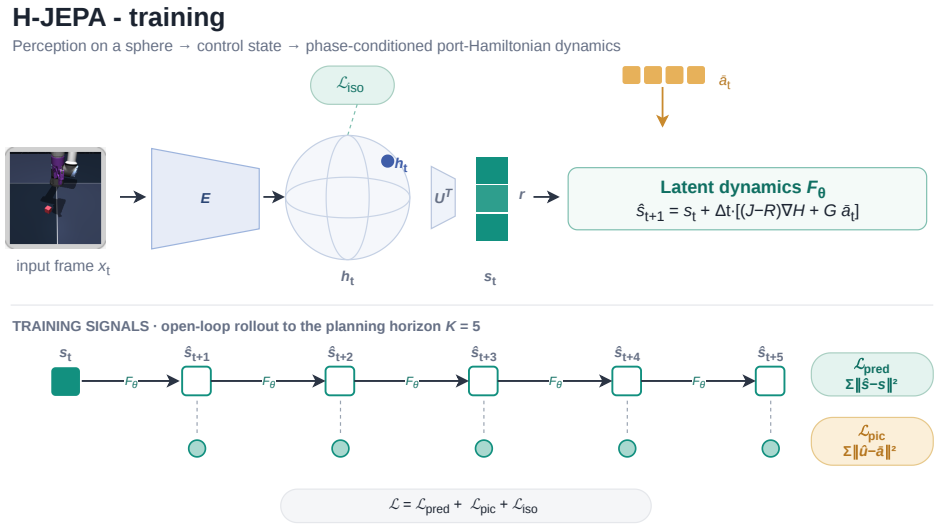


Figure 1: **H-JEPA.** The Bures-Wasserstein prior $\mathcal{L}_{\text{iso}}$ acts only on the spherical code h_t. A fixed orthonormal slice $U^\top$ yields the control state s_t. The phase-conditioned port-Hamiltonian predictor $\mathcal{F}_\theta$ is supervised open-loop ($\mathcal{L}_{\text{pred}}$), PIC reads actions back through the port transpose ($\mathcal{L}_{\text{PIC}}$), and planning scores s alone.

B Derivation of the PIC identity

Fix a rollout step i and abbreviate $c = \hat{c}_{t+i}$, $s = \hat{s}_{t+i}$, $v = \hat{v}_{t+i}$, and $\bar{a} = \bar{a}_{t+i}$. The model's one-step prediction from the rollout state is

$$\hat{s}_{t+i+1} = s + \Delta t\Big[\big(J - R_\theta(c)\big)\nabla_s H_\theta(s,v) + G_\theta(c)\,\bar{a}\Big]. \tag{13}$$

Substituting the readout of equation 9 and using $G_\theta(c)^\top G_\theta(c) = I$,

$$\hat{u}_{t+i} - \bar{a} = G_\theta(c)^\top\Big[\tfrac{s_{t+i+1}-s}{\Delta t} - \big(J - R_\theta(c)\big)\nabla_s H_\theta(s,v)\Big] - G_\theta(c)^\top G_\theta(c)\,\bar{a} \tag{14}$$

$$= \tfrac{1}{\Delta t}\,G_\theta(c)^\top\Big[s_{t+i+1} - s - \Delta t\big(J - R_\theta(c)\big)\nabla_s H_\theta(s,v) - \Delta t\,G_\theta(c)\,\bar{a}\Big] \tag{15}$$

$$= \tfrac{1}{\Delta t}\,G_\theta(c)^\top\big(s_{t+i+1} - \hat{s}_{t+i+1}\big) = \tfrac{1}{\Delta t}\,G_\theta(c)^\top\delta_{t+i+1}, \tag{16}$$

which is equation 11. The PIC residual is the projection of the open-loop state error onto the port subspace, scaled by $1/\Delta t$. Summing $\frac{1}{r}\|\delta\|_2^2$ and $\frac{1}{d_a \Delta t^2}\|G^\top \delta\|_2^2$ over the same steps gives the quadratic form in equation 12, which weights error along the d_a port directions by a factor $1 + r/(d_a \Delta t^2)$ relative to the remaining $r - d_a$ directions. With $\Delta t = 1$ this factor is 17 on Two-Room, 33 on Reacher, 97 on PushT at $r = 192$, and 7.4 on OGB-Cube.

The identity does not remove the action readout. On a transition that the model predicts exactly, the transpose map returns $\hat{u} = \bar{a}$. Off such transitions the readout is an orthogonal projection of the residual displacement, so it never amplifies noise, its magnitude is bounded by the transition energy, and $\mathcal{L}_{\mathrm{PIC}}$ is of order one at initialization in normalized units. The identity also explains the control in §4.1, where the readout uses a fixed random orthonormal $Q^\top$ in place of $G_\theta(c)^\top$ while the forward step keeps G_θ. The same substitution now gives

$$\hat{u}_{t+i} - \bar{a} = \tfrac{1}{\Delta t}\, Q^\top \delta_{t+i+1} + \big(Q^\top G_\theta(c) - I\big)\bar{a}. \tag{17}$$

The first term is again a projected rollout error. The second is new and vanishes only if $G_\theta(c) = Q$ for every visited phase, so the untied loss pulls the port toward a fixed frame while $\mathcal{L}_{\mathrm{pred}}$ asks it to follow the state. With the tied readout this term is identically zero and the port is left free. The control recovers about half of the gain (75.0 to 83.7, against 93.3 with the tied readout), and the tie supplies the rest. Weighting error along the directions in which actions enter is qualitatively related to the controllability side of balanced realizations (Moore, 1981), with predictive accuracy playing the analogous role for observability.

C Default configuration and training details

Table 7: Default H-JEPA configuration. Task-specific entries are shown explicitly for rank, action dimension, and checkpoint epoch.

Quantity	Value
Perceptual dimension D	192
Encoder	ViT-T/14 from scratch, MLP projector $192 \to 2048 \to 192$
Output normalization	affine-free LayerNorm, $\|h\| = \sqrt{D}$, per sample
Projection U	fixed random orthonormal, seeded QR
History and phase	2 frames, $c = [s, v]$ with $v = s - s_{\mathrm{prev}}$, left-pad $v_0 = 0$
Net action dimension d_a	2 on Two-Room, Reacher, PushT, 5 on OGB-Cube
Port constraint	Stiefel, $G^\top G = I$ via differentiable QR
Dissipation	$R = LL^\top$ with lower-triangular L, seeded so $R \approx (\rho_{\mathrm{init}}/r)I$, $\rho_{\mathrm{init}} = 10^{-2}$
Dynamics MLPs	depth 3, width 256, GELU, no dropout
H_θ head initialization	$\mathcal{N}(0, 10^{-3})$ so passive drift starts near zero
Step size Δt	1.0
Horizons	$K_p = 5$ (equal to the CEM horizon), $K_{\mathrm{PIC}} = 5$
Window length T	7
Optimizer	AdamW, lr 5×10^{-5}, weight decay 10^{-3}, bf16, grad clipping
Training length and seeds	10 epochs on every task, checkpoint epoch 8/6/10/3 for Two-Room/Reacher/PushT/OGB-Cube selected on a held-out pilot seed, reported seeds $\{3072, 7, 11\}$
Baseline protocol	Delta-JEPA evaluation protocol (Zhang et al., 2026), with 50 training epochs for Delta-JEPA
CEM	horizon 5, receding 5, 300 samples, 30 iters PushT, 10 elsewhere
Evaluation	$N = 500$ main, lower-budget ablations use $N = 50$ or $N = 300$ as stated
Diagnostic probe	linear head on detached displacements, logging only

Table 7 lists the defaults. The three losses are summed without weights, which places PIC and prediction on comparable scales in normalized action and state units. For context, Zhang et al. (2026) sweep the weight of their learned action decoder in the same action units and report collapse below 0.1 with a broad workable range from 1 to 50. The two losses are not equivalent, so we treat this as context and not as hyperparameter transfer. The readout itself is two matrix products plus one thin QR factorization per step, with no linear solve and no regularizer, run under bf16 autocast with the factorization in a small fp32 island, as is standard for spectral operations.

Table 8: Physical latent probing on PushT (ρ is the Pearson correlation).

Property	Method	Linear MSE ↓	Linear ρ ↑	MLP MSE ↓	MLP ρ ↑
Agent location	H-JEPA	0.028 ± 0.003	0.986	0.008 ± 0.003	0.996
	LeWM	0.062 ± 0.012	0.968	0.017 ± 0.006	0.992
	Sub-JEPA	0.065 ± 0.009	0.966	0.016 ± 0.004	0.992
Block location	H-JEPA	0.152 ± 0.025	0.926	0.049 ± 0.024	0.979
	LeWM	0.038 ± 0.011	0.982	0.024 ± 0.014	0.990
	Sub-JEPA	0.033 ± 0.009	0.984	0.017 ± 0.007	0.992
Block angle	H-JEPA	0.331 ± 0.053	0.822	0.199 ± 0.024	0.896
	LeWM	0.199 ± 0.019	0.898	0.108 ± 0.025	0.944
	Sub-JEPA	0.201 ± 0.013	0.895	0.072 ± 0.015	0.965

Table 9: Physical latent probing on Reacher (ρ is the Pearson correlation).

Property	Method	Linear MSE ↓	Linear ρ ↑	MLP MSE ↓	MLP ρ ↑
Finger position	H-JEPA	0.002 ± 0.000	0.999	0.000 ± 0.000	1.000
	LeWM	0.002 ± 0.000	0.999	0.000 ± 0.000	1.000
	Sub-JEPA	0.002 ± 0.000	0.999	0.000 ± 0.000	1.000
Joint position	H-JEPA	0.601 ± 0.125	0.588	0.742 ± 0.111	0.562
	LeWM	0.636 ± 0.120	0.573	0.738 ± 0.116	0.550
	Sub-JEPA	0.639 ± 0.125	0.575	0.726 ± 0.129	0.559

D FULL PHYSICAL PROBING TABLES

The protocol follows Zhang et al. (2026) and Maes et al. (2026): 20,000 frames, trajectory-level train and test split, targets z-scored on the train split, an OLS linear probe and a 2×256 ReLU MLP probe trained for 100 epochs, and three probe seeds governing MLP initialization and minibatch order (the OLS fit is deterministic given the split), reported as mean $\pm$ std. H-JEPA probes the control state $s = U^\top h$ at the task-specific converged checkpoints defined in §4 (PushT seed 11 at r=192, Reacher seed 3072 at r=64, OGB-Cube seed 3072 at r=32), and LeWM and Sub-JEPA use their official projected 192-d embeddings. Angular and quaternion targets are evaluated in raw coordinates, so wrap-around and sign ambiguity depress correlations for all methods alike, and those rows should be read comparatively. Tables 8, 9, and 10 report PushT, Reacher, and OGB-Cube in turn.

E INTERPRETING THE PORT-HAMILTONIAN PREDICTOR

The port-Hamiltonian parameterization is introduced as an inductive bias rather than an assumption that the learned coordinates recover physical positions, momenta, or energies. We therefore ask a more limited question: *after training for planning, does the predictor use its structured components in a meaningful and task-dependent way?* We audit the best planning checkpoint for each benchmark on 24,576 held-out expert transitions, evaluating the learned fields only on encoded states encountered in the data.

The imposed structure is realized numerically. For each transition we decompose the predicted velocity into

$$\dot{s} = \underbrace{\big(J - R(c)\big)\nabla_s H(s,v)}_{\text{passive flow}} + \underbrace{G(c)\bar{a}}_{\text{control flow}} . \tag{18}$$

The Stiefel constraint is satisfied to numerical precision on all three tasks, $\|G^\top G - I\|_\infty \approx 4.8 \times 10^{-7}$, while $R(c) = L(c)L(c)^\top$ remains positive semidefinite by construction. Thus, the analysis below probes the learned fields within the intended decomposition and not artifacts of constraint violation.

The balance of passive and controlled dynamics is task dependent. Table 11 shows that the predictor does not use the same decomposition across environments. PushT is strongly dominated

Table 10: Physical latent probing on OGB-Cube (ρ is the Pearson correlation).

Property	Method	Linear MSE ↓	Linear ρ ↑	MLP MSE ↓	MLP ρ ↑
Joint position	H-JEPA	0.445 ± 0.044	0.670	0.554 ± 0.056	0.685
	LeWM	0.579 ± 0.060	0.645	0.775 ± 0.109	0.647
	Sub-JEPA	0.492 ± 0.067	0.676	0.557 ± 0.059	0.675
Joint velocity	H-JEPA	0.740 ± 0.027	0.432	0.713 ± 0.039	0.541
	LeWM	1.069 ± 0.038	0.088	2.674 ± 0.236	0.050
	Sub-JEPA	0.988 ± 0.032	0.203	1.595 ± 0.226	0.101
End-effector position	H-JEPA	0.007 ± 0.000	0.997	0.002 ± 0.000	0.999
	LeWM	0.027 ± 0.003	0.987	0.020 ± 0.001	0.991
	Sub-JEPA	0.025 ± 0.002	0.988	0.014 ± 0.000	0.993
End-effector yaw	H-JEPA	1.315 ± 0.291	-0.124	2.133 ± 0.448	-0.113
	LeWM	2.046 ± 0.483	-0.161	2.198 ± 0.370	-0.143
	Sub-JEPA	1.575 ± 0.411	0.008	2.115 ± 0.380	-0.078
Block position	H-JEPA	0.023 ± 0.001	0.989	0.009 ± 0.001	0.996
	LeWM	0.011 ± 0.001	0.995	0.010 ± 0.001	0.995
	Sub-JEPA	0.053 ± 0.005	0.975	0.010 ± 0.001	0.995
Block quaternion	H-JEPA	1.352 ± 0.439	0.138	1.753 ± 0.421	0.103
	LeWM	1.822 ± 0.403	-0.010	2.512 ± 0.505	-0.035
	Sub-JEPA	1.662 ± 0.434	0.021	2.273 ± 0.358	-0.010
Block yaw	H-JEPA	1.142 ± 0.233	0.184	1.804 ± 0.222	0.024
	LeWM	1.779 ± 0.448	-0.025	1.856 ± 0.368	-0.050
	Sub-JEPA	1.313 ± 0.270	0.208	1.589 ± 0.558	0.161

Table 11: **Learned port-Hamiltonian structure on held-out transitions.** The relative contribution of dissipation, passive dynamics, and direct control changes markedly across tasks.

Task	$\frac{\|R\nabla H\|}{\|J\nabla H\|}$	$\frac{\|G\bar{a}\|}{\|\text{passive}\|}$	$\frac{\text{tr}(G^\top RG)}{\text{tr}(R)}$	$\text{corr}(\text{tr}\, R, \|v\|)$
PushT	4.35	0.13	0.001	0.48
Reacher	1.16	0.56	0.008	0.24
OGB-Cube	1.11	1.53	0.184	0.13

by passive, dissipative flow: $\|R\nabla H\|/\|J\nabla H\| = 4.35$, while the direct control flow is only 0.13 times the passive flow on expert trajectories. Reacher is more balanced, and OGB-Cube becomes control dominated, with $\|G\bar{a}\|/\|\text{passive}\| = 1.53$. The overlap $\text{tr}(G^\top RG)/\,\text{tr}(R)$ is nearly zero on PushT but substantially larger on OGB-Cube, indicating that the learned dissipative and actuated subspaces themselves adapt to the task.

PushT provides the clearest physical interpretation. Its learned dissipation increases with latent speed ($\text{corr}(\text{tr}\, R, \|v\|) = 0.48$), and the dissipative component dominates the conservative component, consistent with a representation organized around the strongly dissipative contact dynamics of pushing. OGB-Cube exhibits the opposite regime: direct actuation is larger than the passive field and actuation overlaps more strongly with the learned dissipative subspace. We emphasize that these quantities are measured in learned latent coordinates and should be interpreted as structural signatures, not estimates of physical coefficients in SI units.

When the task permits it, the learned fields align with physical change. To test whether the decomposition contains interpretable information beyond its relative magnitudes, we fit linear ridge probes from each field to the corresponding physical state increment Δy. Results are shown in Table 12. On PushT, the passive field, control field, and Hamiltonian gradient each explain approximately 88 to 90 percent of the variance in the physical increment. The control state itself predicts the physical state with $R^2 = 0.95$. Thus, on this task the latent decomposition is not only structurally valid but closely aligned with observable physical evolution.

Table 12: **Linear physical alignment of the learned predictor.** R^2 is measured on held-out transitions using ridge regression.

Task	Passive $\to \Delta y$	$G\bar{a} \to \Delta y$	$\nabla H \to \Delta y$	$s \to y$
PushT	0.881	0.875	0.899	0.953
Reacher	0.351	0.033	0.291	0.514
OGB-Cube	0.268	0.323	0.312	0.561

The same factor-wise interpretation is weaker on Reacher and OGB-Cube. This does not imply that the learned state is non-physical: complementary probes recover finger position in Reacher with correlation 0.999, and end-effector and block position in OGB-Cube with correlations 0.997 and 0.989, respectively. Rather, these tasks appear to represent the variables useful for planning more holistically in s, without requiring each of the H, R, and G components to be linearly aligned with a privileged physical coordinate.

Interpretation. These results suggest two levels at which the structured predictor is useful. First, the port-Hamiltonian parameterization provides a calibrated decomposition into passive, dissipative, and directly controlled change whose relative use adapts across tasks. Second, when the underlying task admits a simple physical factorization, as in PushT, the learned fields themselves become strongly interpretable in terms of physical evolution. Successful planning does not require this stronger property: on Reacher and OGB-Cube, control-relevant physical information remains clearly present in the planning state even when it is distributed across the nonlinear predictor. We therefore view the port-Hamiltonian structure primarily as an inductive bias that organizes latent dynamics, with physical interpretability emerging where supported by the task rather than being imposed by construction.

F History mechanism ablation

An earlier variant used a three-frame history fused by a learned mixer with LayerNorm, $c = \mathrm{LN}(W_{\mathrm{hist}}[s, \Delta_1, \Delta_2])$. After training, about 90 percent of the mixer's Frobenius energy concentrated on the latest state, and zeroing the difference channels barely changed one-step predictions (cosine similarity about 0.99), so the mixer was learning to ignore the history it was built to fuse. The current parameter-free phase $c = [s, v]$ with a two-frame window removes the mixer and its normalization entirely, keeps s a single-frame representation, and leaves the planner's metric untouched, and it is the variant used in all selected recipes.

G Geometry of the inherited control state

This appendix makes the inheritance statement of §3.2 exact and compares it with trained models.

Target covariance of the state. Let $e = \mathbf{1}/\sqrt{D}$. LN_0 gives $e^\top h = 0$ for every sample, so $\mathrm{Cov}(h)\, e = 0$ for any encoder, and the target of equation 3 is $P = I - ee^\top$. The exact covariance of a uniform distribution on the radius-$\sqrt{D}$ sphere inside the hyperplane is $\frac{D}{D-1}P$, which differs from P by 0.5 percent at $D = 192$, and we use the unit target as a fixed convention. If $\mathrm{Cov}(h) = P$, then by equation 2

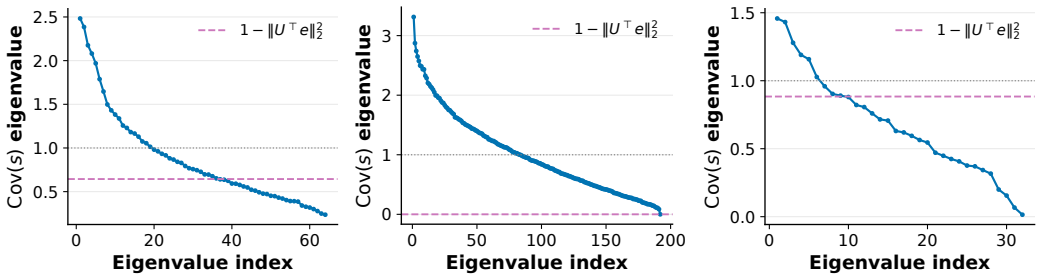
$$\mathrm{Cov}(s) = U^\top (I - ee^\top) U = I_r - ww^\top, \qquad w = U^\top e. \tag{19}$$

This matrix has eigenvalue 1 with multiplicity $r - 1$ and one exceptional eigenvalue $1 - \|w\|_2^2$ along w. For U drawn uniformly among orthonormal frames, $\|w\|_2^2$ follows a $\mathrm{Beta}\big(\frac{r}{2}, \frac{D-r}{2}\big)$ distribution with mean r/D. When $r = D$, U is an orthogonal matrix, $\|w\|_2 = 1$, and the exceptional eigenvalue is exactly zero: the state then has one null direction, the image of the centering direction.

What trained models show. Table 13 and Figure 2 support three statements. First, the measured $\|U^\top e\|_2^2$ is close to r/D for each seeded projection, as expected. Second, the structural prediction at $r = D$ is confirmed: the smallest eigenvalue of $\mathrm{Cov}(s)$ is 1.3×10^{-6}, so the full-rank PushT state lives in a hyperplane of dimension $D - 1$. Third, for $r < D$ every state direction retains nonzero variance, but the spectrum is spread around one and is not flat, and the smallest eigenvalue on OGB-Cube is well below the Marchenko-Pastur range. Part of this spread is a finite-sample effect, as the

Table 13: Measured geometry of the control state on $n = 512$ encoded frames from the selected checkpoints. The Marchenko-Pastur (MP) interval is the range of sample eigenvalues that an exactly isotropic state would show at this n and r (Marčenko & Pastur, 1967).

			exceptional	measured eigenvalues of $\mathrm{Cov}(s)$			
Model	$\|U^\top e\|_2^2$	r/D	$1 - \|U^\top e\|_2^2$	min	median	max	MP interval
PushT, $r = 64$	0.356	0.333	0.644	0.236	0.713	2.48	[0.42, 1.83]
PushT, $r = 192$	1.000	1.000	0.000	1.3×10^{-6}	0.877	3.31	[0.15, 2.60]
OGB-Cube, $r = 32$	0.116	0.167	0.884	0.015	0.625	1.46	[0.56, 1.56]



(a) PushT, $r = 64$ **(b)** PushT, $r = 192$ **(c)** OGB-Cube, $r = 32$

Figure 2: **Eigenvalues of $\mathrm{Cov}(s)$ for the selected checkpoints.** The dashed line marks the exceptional eigenvalue $1 - \|U^\top e\|_2^2$ of the target covariance and the dotted line marks one. At $r = D = 192$ the predicted null direction appears as the last eigenvalue.

MP intervals show, and part is residual anisotropy of the code, whose own covariance eigenvalues reach about 2.75 in these checkpoints. The smallest measured eigenvalue lies below the exceptional value of the target in both $r < D$ models, so the tail of the spectrum is set by this residual anisotropy and not by the centering direction. The tail is lowest on OGB-Cube, whose selected checkpoint is from epoch 3. We therefore describe the state as inheriting a full-rank, well-scaled covariance from the code, which is exactly isotropic only at the prior's optimum.

H Anti-collapse penalties near vanishing variance

Consider one covariance direction with variance λ and standard deviation $\sigma = \sqrt{\lambda}$, and write the samples along it as $z_j = \sigma x_j$ with standardized x_j. For any penalty ℓ, the chain rule gives

$$\frac{\partial \ell}{\partial \sigma} = 2\sigma \frac{\partial \ell}{\partial \lambda}, \tag{20}$$

so a penalty whose eigenvalue derivative stays bounded as $\lambda \to 0$ exerts a force on σ that vanishes linearly, and only a derivative that diverges like $\lambda^{-1/2}$ keeps a constant force. The three penalties of equation 4 fall on the two sides of this line, as Figure 3 shows.

Bures-Wasserstein. $\ell = (\sqrt{\lambda} - 1)^2$ has $\partial_\lambda \ell = 1 - \lambda^{-1/2}$, which diverges at exactly the compensating rate, and $\partial_\sigma \ell = 2(\sigma - 1) \to -2$.

Euclidean covariance penalty. $\ell = (\lambda - 1)^2$ has $\partial_\lambda \ell = 2(\lambda - 1) \to -2$, bounded, and $\partial_\sigma \ell = 4\sigma(\sigma^2 - 1) \to 0$.

SIGReg. The Epps-Pulley statistic (EPPS & PULLEY, 1983) compares the empirical characteristic function $\varphi_n(\tau) = \frac{1}{n}\sum_j \exp(\mathrm{i}\tau\sigma x_j)$ with that of a standard Gaussian under a weight ω, $T(\sigma) = n \int |\varphi_n(\tau) - e^{-\tau^2/2}|^2 \, \omega(\tau)\, d\tau$. Expanding at small σ with standardized x_j gives $\varphi_n(\tau) = 1 - \frac{1}{2}\tau^2\sigma^2 + O(\sigma^3)$, where the cubic term is purely imaginary, hence

$$T(\sigma) = T(0) - \kappa\,\sigma^2 + O(\sigma^4), \qquad \kappa = n \int \tau^2 \big(1 - e^{-\tau^2/2}\big)\omega(\tau)\, d\tau > 0. \tag{21}$$

The statistic is bounded, since $|\varphi_n(\tau) - e^{-\tau^2/2}| \leq 2$, its eigenvalue derivative tends to the constant $-\kappa$, and its force on σ is $-2\kappa\sigma \to 0$. A collapsing direction therefore sees a saturated SIGReg score and a vanishing restoring force, as we observed in training.

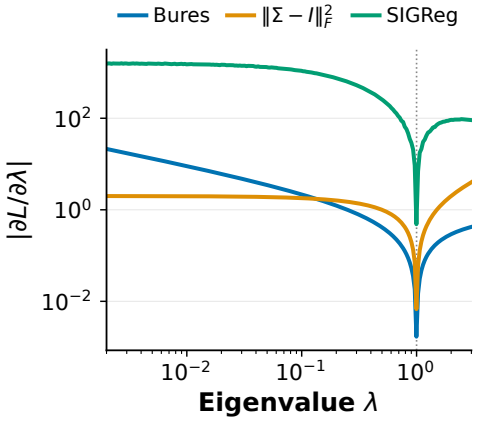
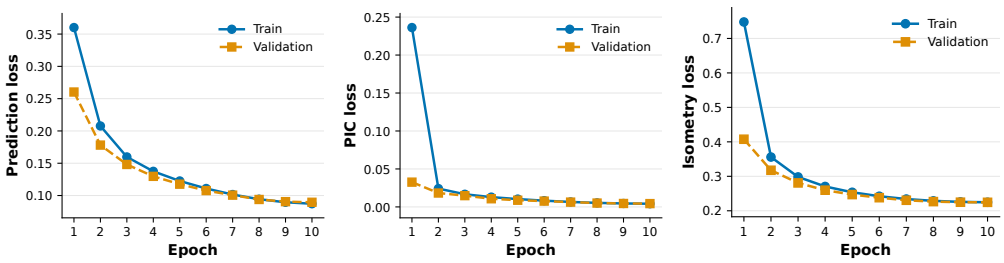


Figure 3: **Eigenvalue derivatives of the three penalties on synthetic data.** The plot shows $|\partial\ell/\partial\lambda|$ against λ. Bures-Wasserstein rises as $\lambda^{-1/2}$ when $\lambda \to 0$, while the Euclidean penalty and SIGReg flatten to constants. By equation 20, a flat curve means a force on the standard deviation that vanishes like $\sqrt{\lambda}$, and the $\lambda^{-1/2}$ rise means a constant force. Vertical offsets reflect the native normalization of each statistic and are not comparable across curves. The comparison is between slopes.

I Training behavior

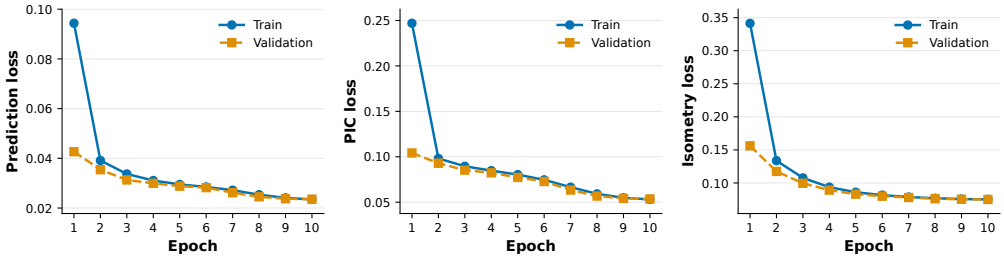
All H-JEPA models were trained end-to-end for 10 epochs on a single NVIDIA RTX A5000 GPU with the optimization settings of Appendix C. A held-out pilot seed, excluded from the three-seed benchmark average, was used once to choose which epoch's checkpoint to evaluate on each task. We then fixed this choice at epoch 8 for Two-Room, 6 for Reacher, 10 for PushT, and 3 for OGB-Cube for every reported seed. For comparison, Delta-JEPA is trained for 50 epochs under its evaluation protocol (Zhang et al., 2026). Figures 4 and 5 show the available extended traces for PushT and OGB-Cube, including prediction, PIC, and Bures-Wasserstein isotropy on both training and validation data.

The curves remain closely matched between training and validation. PushT improves steadily over all 10 epochs. OGB-Cube reaches its useful regime within the first three epochs, after which the representation-level losses continue to decrease without a corresponding need for a later checkpoint. The traces also show that all three objectives can be optimized jointly without loss-specific warm-up schedules or alternating optimization.



(a) Prediction **(b)** Port-inverse consistency **(c)** Isotropy

Figure 4: **H-JEPA training on PushT.** Training and validation losses across the 10-epoch run for **(a)** open-loop latent prediction, **(b)** port-inverse consistency (PIC), and **(c)** the Bures-Wasserstein isotropy objective. All objectives decrease smoothly, with closely tracking training and validation curves.



(a) Prediction **(b)** Port-inverse consistency **(c)** Isotropy

Figure 5: **H-JEPA training on OGB-Cube.** Training and validation losses for **(a)** open-loop latent prediction, **(b)** PIC, and **(c)** isotropy. All three objectives fall rapidly during the first epochs and subsequently improve smoothly, with little separation between training and validation curves. The benchmark model uses the epoch-3 OGB-Cube checkpoint selected in §4. The extended trace shows that lower representation-level losses after convergence are not required for the reported planning result.