Beyond Isotropy in JEPAs: Hamiltonian Geometry and Symplectic Prediction

Robert Jenkinson Alvarez

Abstract

JEPAs often regularize one-view embeddings toward an isotropic Gaussian, implicitly baking Euclidean symmetry into the representation. We show that this is not merely a benign default. For a known structured downstream geometry $H \succ 0$, the minimax and maximum-entropy covariance under a Hamiltonian energy budget is $(c/d)H^{-1}$, and Euclidean isotropy incurs a closed-form price of isotropy. More importantly, when the downstream geometry is unknown, no geometry-independent fixed marginal target is canonical: every fixed covariance shape can be maximally misaligned for some structured geometry. We further show that even oracle one-view marginals do not identify the JEPA view-to-view predictive coupling. Within continuous linear phase-space flows, stationarity of the lifted oracle law, momentum reversal, and fixed-unit kinematics uniquely select Hamiltonian dynamics. These results suggest that the structural bias in JEPAs should enter the cross-view coupling rather than a fixed encoder marginal. We instantiate this principle with **HamJEPA**, which encodes each view as a phase-space state (q, p) and predicts view-to-view transitions with a learned Hamiltonian leapfrog map, while non-isotropic scale and spectral floors discourage collapse. In a deliberately headless token protocol, HamJEPA improves over SIGReg on CIFAR-100 by $+4.89$ kNN@20 and $+3.52$ linear-probe points at 30 epochs, and, over two training seeds at 80 epochs, by $+7.24$ kNN@20 and $+9.67$ linear-probe points relative to the full 2048-dimensional SIGReg representation. Matched non-symplectic and non-reversible controls retain the gain, so the experiments do not identify either property as individually necessary. On ImageNet-100, HamJEPA-q improves by $+4.82$ kNN@20 and $+7.52$ linear-probe points at 45 epochs.

1 Introduction

Joint Embedding Predictive Architectures (JEPAs) learn representations by predicting one embedding from another. For a fixed predictive loss, downstream performance depends on embedding geometry: scale, anisotropy, and collapse. We study objectives of the form $\min_\theta \; \mathcal{L}_{\text{JEPA}}(\theta) + \lambda \, \mathcal{L}_{\text{reg}}(Z_\theta)$, where $\mathcal{L}_{\text{reg}}$ regularizes the embedding distribution induced by the encoder.

A standard target is an *isotropic Gaussian* embedding distribution. LeJEPA motivates this choice under a "no downstream structure" viewpoint and enforces it with Sketched Isotropic Gaussian Regularization (SIGReg) [3]. Isotropy is a symmetry assumption: it treats all embedding directions as exchangeable. In many settings (patch-token grids, sequences, graphs), the index set carries known structure that induces preferred correlations and invariances; enforcing Euclidean isotropy can misalign the representation with this geometry.

This paper makes three linked contributions. First, for a fixed structured downstream geometry H, we derive the minimax and maximum-entropy covariance $(c/d)H^{-1}$ and the price paid by Euclidean isotropy. Second, we show that this oracle result should not be interpreted as a new universal marginal target: when H is unknown, no geometry-independent fixed covariance shape is canonical, and even oracle marginals do not identify predictive coupling. Under an explicit stationary Gaussian

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

lift, momentum reversibility and fixed-unit kinematics then characterize a unique linear Hamiltonian coupling. Third, we instantiate the resulting design principle with a phase-space JEPA (*HamJEPA*) that places Hamiltonian structure in the cross-view predictor through a symplectic leapfrog map while using non-isotropic scale and spectral floors to discourage collapse.

Code: GitHub: HamJEPA-HamSIGReg. Source provenance: Appendix H.

1.1 Positioning relative to prior regularizers and dynamics priors

LeJEPA regularizes Euclidean embeddings toward an isotropic Gaussian using SIGReg, a sliced characteristic-function (CF) matching objective [3]. KerJEPA broadens this marginal-discrepancy viewpoint with kernel-based discrepancies [25]. HamJEPA departs from this line by asking when Euclidean isotropy is misaligned with structured task geometry and by placing Hamiltonian structure in the predictive map rather than prescribing a Euclidean-isotropic encoder marginal.

Several SSL objectives prevent collapse through low-order or spectral statistics, including Barlow Twins, VICReg, MCR^2, and coding-rate-regularized self-distillation [4, 21–23]. HamJEPA uses related scale and spectral non-degeneracy controls, but couples them to a phase-space JEPA with a symplectic predictor; we do not claim these anti-collapse penalties are novel in isolation.

Hamiltonian Neural Networks, Hamiltonian Generative Networks, and Symplectic ODE-Net impose energy-based or symplectic structure on learned dynamics, typically for temporal or control data [9, 18, 24]. HamJEPA uses the same geometric bias in a different role: a predictor prior for paired augmentations, not a physics model of observed trajectories.

2 Setup

Data, views, and randomness. Let $x \sim \mathcal{D}$ be a data sample and let $\tau \sim \mathcal{A}$ be a random augmentation. A *view* is $v := \tau(x)$. We sample paired views of the same sample:

$$v_a = \tau_a(x), \qquad v_b = \tau_b(x). \tag{1}$$

We use two representation parameterizations:

(A) Euclidean embedding. An encoder $f_\theta : \mathcal{X} \to \mathbb{R}^d$ maps a view to an embedding random variable

$$Z \;:=\; f_\theta(V) \in \mathbb{R}^d, \tag{2}$$

where V denotes a generic random view. Assume finite second moments. Let $\mu_\theta := \mathbb{E}[Z]$ and $\Sigma_\theta := \mathrm{Cov}(Z) \succeq 0$. We write centered embeddings as $\widetilde{Z} := Z - \mu_\theta$ so $\mathbb{E}[\widetilde{Z}] = 0$ and $\mathrm{Cov}(\widetilde{Z}) = \Sigma_\theta$.

(B) Phase-space state (HamJEPA). A phase-space encoder $E_\theta : \mathcal{X} \to \mathbb{R}^{2d_0}$ outputs a state

$$S := E_\theta(V) = (Q, P), \qquad Q, P \in \mathbb{R}^{d_0}. \tag{3}$$

We interpret Q as a *content* coordinate and P as an auxiliary *momentum* coordinate used to model structured view-to-view transitions. We write projections $\pi_q(Q, P) = Q$ and $\pi_p(Q, P) = P$.

Batch (row) convention. For a minibatch of size n, we stack embeddings/states as row vectors: $\mathbf{Z} \in \mathbb{R}^{n \times d}$ has rows $z_i^\top$, and $\mathbf{S} \in \mathbb{R}^{n \times 2d_0}$ has rows $s_i^\top = (q_i^\top, p_i^\top)$. We center batches by subtracting the row-mean, e.g. $\widetilde{\mathbf{Z}} = \mathbf{Z} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\mathbf{Z}$.

Two Hamiltonians (to avoid notation collisions). Section 3 uses an SPD operator $H \succ 0$ to specify a *geometry* on embedding space. HamJEPA uses a *scalar energy* $\mathcal{H}_\phi(q, p)$ generating a symplectic map on phase space. We reserve H for the SPD operator and $\mathcal{H}_\phi$ for the scalar energy.

3 A Theory of Marginal Geometry and Predictive Coupling in JEPAs

This section is pure theory: given a structured notion of task complexity and an energy budget, what covariance/distribution is canonical, and what is the cost of forcing Euclidean isotropy? Throughout this section, fix an SPD operator $H \succ 0$, which defines the structured embedding geometry through the quadratic energy $z^\top H z$; Hamiltonian-isotropic covariance in this geometry has the form αH^{-1}. We work with the centered embedding $\widetilde{Z} := Z - \mathbb{E}[Z]$ and write $\Sigma := \mathrm{Cov}(\widetilde{Z})$.

3.1 Setup: structured tasks and a Hamiltonian budget

We model downstream *linear* tasks by $y = w^\top \widetilde{Z}$ and constrain task complexity by the H^{-1}-unit ball

$$\mathcal{W}_H := \{w \in \mathbb{R}^d : \ w^\top H^{-1} w \le 1\}. \tag{4}$$

We pair this with a representation energy budget in the H-geometry:

$$\mathbb{E}[\widetilde{Z}^\top H \widetilde{Z}] = \operatorname{tr}(H\Sigma) = c, \qquad c > 0. \tag{5}$$

Given $\Sigma \succeq 0$, define the worst-case task variance functional

$$V(\Sigma) := \sup_{w \in \mathcal{W}_H} w^\top \Sigma w. \tag{6}$$

This functional measures worst-case linear-task variance under the stated task class; the results below are not bounds on classification or prediction error. The next lemma gives a convenient spectral form.

Lemma 3.1 (Spectral form of $V(\Sigma)$)**.** *For any $\Sigma \succeq 0$,*

$$V(\Sigma) = \lambda_{\max}\big(H^{1/2}\Sigma H^{1/2}\big). \tag{7}$$

Moreover, if $u_{\max}$ is a unit top eigenvector of $H^{1/2}\Sigma H^{1/2}$, then $w_{\max} := H^{1/2}u_{\max}$ attains the supremum in (6)*.*

Proof. See Appendix C.1.1. □

3.2 Minimax covariance under a Hamiltonian budget

We ask for the covariance that minimizes the worst-case task variance (6) under the H-budget (5).

Theorem 3.2 (Minimax covariance under a Hamiltonian budget)**.** *Fix $H \succ 0$ and $c > 0$. Consider*

$$\min_{\Sigma \succeq 0:\ \operatorname{tr}(H\Sigma) = c} V(\Sigma). \tag{8}$$

The unique minimizer is

$$\Sigma^\star = \frac{c}{d} H^{-1}, \tag{9}$$

and the optimal value is

$$\min_{\Sigma \succeq 0:\ \operatorname{tr}(H\Sigma) = c} V(\Sigma) = \frac{c}{d}. \tag{10}$$

Equivalently, $H^{1/2}\Sigma^\star H^{1/2} = \frac{c}{d} I$.

Proof. See Appendix C.1.2. □

3.3 Maximum entropy: the canonical distribution

The minimax result selects a canonical covariance. A complementary principle selects a canonical *distribution*.

Theorem 3.3 (Maximum entropy under a Hamiltonian energy constraint)**.** *Let Z be an $\mathbb{R}^d$-valued random vector with a density, $\mathbb{E}[Z] = 0$, and finite second moments. Among all such distributions satisfying*

$$\mathbb{E}[Z^\top H Z] = c, \tag{11}$$

the unique maximizer of differential entropy is the Gaussian

$$Z \sim \mathcal{N}\Big(0, \frac{c}{d} H^{-1}\Big). \tag{12}$$

Proof. See Appendix C.1.3. □

3.4 Price of Euclidean isotropy

Many practical objectives encourage Euclidean isotropy, i.e., $\Sigma \propto I$ in the original coordinates. This subsection quantifies the worst-case gap incurred by imposing Euclidean isotropy under the same Hamiltonian budget.

Under the restriction $\Sigma = \sigma^2 I$, the constraint $\mathrm{tr}(H\Sigma) = c$ forces the unique feasible Euclidean-isotropic covariance

$$\Sigma_{\mathrm{iso}} = \frac{c}{\mathrm{tr}(H)}\, I. \tag{13}$$

Theorem 3.4 (Price of Euclidean isotropy)**.** *Fix $H \succ 0$ and $c > 0$. Let $\Sigma^\star$ be as in Theorem 3.2 and let Σ_{iso} be as in* (13)*. Then*

$$V(\Sigma_{\mathrm{iso}}) = \frac{c}{\mathrm{tr}(H)}\, \lambda_{\max}(H), \tag{14}$$

$$\frac{V(\Sigma_{\mathrm{iso}})}{V(\Sigma^\star)} = \rho(H) := \frac{d\, \lambda_{\max}(H)}{\mathrm{tr}(H)}. \tag{15}$$

Moreover $1 \le \rho(H) \le d$, with $\rho(H) = 1$ if and only if $H \propto I$.

Proof. See Appendix C.1.4. □

Proposition 3.5 (Metric isotropy equivalence)**.** *Let $H \succ 0$ and let $Z \in \mathbb{R}^d$ be a random vector with finite second moments and covariance $\Sigma := \mathrm{Cov}(Z)$. Define the H-whitened representation $Z' := H^{1/2} Z$. Then*

$$\mathrm{Cov}(Z') \;=\; H^{1/2} \Sigma H^{1/2}.$$

In particular, for any $\alpha > 0$,

$$\Sigma = \alpha H^{-1} \quad \Longleftrightarrow \quad \mathrm{Cov}(Z') = \alpha I.$$

Proof. See Appendix C.1.5. □

3.5 A quadratic Hamiltonian lift of the oracle structured Gaussian

The structured Gaussian selected by Theorems 3.2 and 3.3 admits a natural lift to a Gibbs law on phase space.

Proposition 3.6 (Quadratic Hamiltonian lift of the oracle covariance)**.** *Fix $H \succ 0$ and $c > 0$, and define the quadratic Hamiltonian*

$$\mathcal{H}_H(q,p) := \tfrac{1}{2} q^\top H q + \tfrac{1}{2}\|p\|_2^2, \qquad q, p \in \mathbb{R}^d. \tag{16}$$

Consider the Gibbs density on phase space

$$\pi_H(q,p) \propto \exp\Big(-\frac{d}{c}\, \mathcal{H}_H(q,p) \Big). \tag{17}$$

Then q and p are independent under π_H, with

$$q \sim \mathcal{N}\Big(0, \frac{c}{d} H^{-1}\Big), \qquad p \sim \mathcal{N}\Big(0, \frac{c}{d} I\Big). \tag{18}$$

Moreover,

$$\mathbb{E}_{\pi_H}[q^\top H q] = c, \qquad \mathbb{E}_{\pi_H}[\|p\|_2^2] = c, \qquad \mathbb{E}_{\pi_H}[\mathcal{H}_H(q,p)] = c. \tag{19}$$

In particular, the q-marginal of π_H is exactly the minimax-optimal and maximum-entropy law from Theorems 3.2 and 3.3.

Proof. See Appendix C.1.6. □

Proposition 3.7 (The oracle family spans the SPD cone)**.** *Fix $c > 0$. As H ranges over SPD operators, the family*

$$\Sigma^\star(H) = \frac{c}{d} H^{-1} \tag{20}$$

ranges over the entire SPD cone. Equivalently, for every $\Sigma \succ 0$, choosing

$$H = \frac{c}{d} \Sigma^{-1} \tag{21}$$

yields $\Sigma^\star(H) = \Sigma$.

Proof. See Appendix C.1.7. □

From the oracle covariance to a practical unknown-H relaxation. Proposition 3.6 shows that when H is known, the oracle covariance $\Sigma^\star = \frac{c}{d}H^{-1}$ is exactly the q-marginal of a quadratic Hamiltonian Gibbs model on phase space. In that quadratic oracle case, the induced dynamics are

$$\dot{q} = p, \qquad \dot{p} = -Hq, \tag{22}$$

so a Hamiltonian predictor is not an unrelated construction, but a natural dynamical realization of the same structured geometry.

This does *not* mean that the practical training mechanism is uniquely determined by Section 3. Rather, the theory should be read as an oracle statement: when H is known, the stated minimax and maximum-entropy problems select $\Sigma^\star = (c/d)H^{-1}$. When H is unknown, Proposition 3.7 shows that the oracle family $\{\Sigma^\star(H) : H \succ 0\}$ spans the entire SPD cone, so the theory no longer singles out one fixed covariance target.

Why not learn another fixed marginal target? The previous results might suggest replacing Euclidean isotropy by a better fixed target covariance. The next result shows why this is brittle when downstream geometry is unknown.

Theorem 3.8 (No universal fixed marginal target)**.** *Assume $d \geq 2$. Fix any SPD target shape $M \succ 0$ independent of the downstream geometry. Under geometry $H \succ 0$, the budget-feasible covariance with shape M is*

$$\Sigma_M(H) := \frac{c}{\mathrm{tr}(HM)}M.$$

Its regret relative to the oracle covariance $\Sigma^\star(H) = (c/d)H^{-1}$ is

$$\frac{V_H(\Sigma_M(H))}{V_H(\Sigma^\star(H))} = \frac{d\,\lambda_{\max}(H^{1/2}MH^{1/2})}{\mathrm{tr}(HM)}.$$

This quantity equals 1 iff $M \propto H^{-1}$, and for every fixed $M \succ 0$,

$$\sup_{H \succ 0} \frac{V_H(\Sigma_M(H))}{V_H(\Sigma^\star(H))} = d.$$

Thus no geometry-independent fixed covariance shape is canonical for all structured downstream geometries.

Proof. See Appendix C.1.8. □

Proposition 3.9 (Oracle marginals do not identify the predictive coupling)**.** *Let $Z_a, Z_b \in \mathbb{R}^d$ be centered jointly Gaussian random vectors with $\mathrm{Cov}(Z_a) = \mathrm{Cov}(Z_b) = \Sigma \succ 0$ and cross-covariance $\mathrm{Cov}(Z_b, Z_a) = C$. Then*

$$\mathbb{E}[Z_b \mid Z_a] = C\Sigma^{-1}Z_a.$$

Therefore, the same oracle marginal covariance Σ admits a continuum of different predictive maps as the admissible cross-covariance C varies. Marginal matching alone cannot determine the JEPA view-to-view prediction problem.

Proof. See Appendix C.1.9. □

3.6 A canonical reversible coupling of the oracle law

The preceding non-identification result leaves open which additional assumptions select a coupling. We now specify a linear phase-space oracle; the assumptions below are additional modeling choices, rather than consequences of marginal matching.

Theorem 3.10 (Canonical linear oracle lift)**.** *Fix $H \succ 0$, $\alpha = c/d > 0$, and*

$$\Gamma_H = \alpha \begin{pmatrix} H^{-1} & 0 \\ 0 & I \end{pmatrix}, \qquad \mathsf{R} = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix}.$$

Let F_t be a continuous one-parameter group of linear maps on $\mathbb{R}^{2d}$. Suppose it preserves $\mathcal{N}(0, \Gamma_H)$, satisfies momentum reversibility $\mathsf{R}F_t\mathsf{R} = F_{-t}$, and obeys the fixed-unit kinematics $\dot{Q} = P$. Then, uniquely,

$$F_t = e^{tL_H}, \qquad L_H = \begin{pmatrix} 0 & I \\ -H & 0 \end{pmatrix}, \qquad \dot{Q} = P, \quad \dot{P} = -HQ. \tag{23}$$

Proof. See Appendix D.1. □

Thus the quadratic Hamiltonian from Proposition 3.6 is characterized within this explicit oracle class: symplecticity follows from the assumptions. Stationarity and momentum reversal provide a sufficient latent construction for equal, exchangeable observable view marginals; those observable properties alone do not imply the latent assumptions.

Corollary 3.11 (Observable coupling and geometry)**.** *Under the stationary law in Theorem 3.10,*

$$\begin{aligned} Q_t \mid Q_0 = q &\sim \mathcal{N}\Big(C_t q,\, \alpha H^{-1/2} \sin^2(tH^{1/2}) H^{-1/2}\Big), \\ C_t = \cos(tH^{1/2}), &\qquad M(t) := \mathrm{Cov}(Q_t, Q_0)\,\mathrm{Cov}(Q_0)^{-1} = C_t, \\ H = -M''(0). & \end{aligned} \tag{24}$$

The pair (Q_0, Q_t) is exchangeable.

Proof. See Appendix D.2. □

Invertible full-state dynamics therefore induce a generally stochastic observable transition after momentum is marginalized. The curvature identity describes the oracle coupling near time zero; it does not establish recovery of an unknown downstream metric by a single-horizon training loss.

Proposition 3.12 (Exact quadratic leapfrog coupling)**.** *Let $h > 0$ satisfy $h^2 \lambda_{\max}(H) < 4$, and define*

$$R_h = I - \tfrac{h^2}{4} H, \qquad \Gamma_{H,h} = \alpha\, \mathrm{diag}(H^{-1}, R_h).$$

Quadratic leapfrog for $\mathcal{H}_H$ exactly preserves $\mathcal{N}(0, \Gamma_{H,h})$. Under this law its normalized K-step observable cross-covariance is

$$M_{K,h} := \mathrm{Cov}(Q_K, Q_0)\,\mathrm{Cov}(Q_0)^{-1} = T_K\Big(I - \tfrac{h^2}{2} H\Big), \tag{25}$$

where T_K is the Chebyshev polynomial of the first kind.

Proof. See Appendix D.3. □

This exact quadratic shadow law retains the Section 3 content marginal and corrects the momentum covariance by R_h; finite-step leapfrog does not preserve the original full oracle lift exactly. The filter converges to $\cos(tH^{1/2})$ when $h \to 0$ and $Kh \to t$. For a sparse grid or graph operator H, its degree-K polynomial also gives finite-hop locality, as detailed in the proof. These exact Gaussian and polynomial statements concern the quadratic oracle. Section 4 uses a learned nonlinear potential and encoder-side penalties without requiring the encoder law to equal either oracle law. The characterization is structural, not a downstream performance ordering; Appendix D.4 gives an explicit oracle-incompatible control with identical Bayes performance on a task.

Together, Theorem 3.8 and Proposition 3.9 show that the unknown-geometry JEPA problem should not be reduced to choosing a single fixed encoder marginal. This motivates specifying additional structural bias in the view-to-view coupling. HamJEPA instantiates this principle by using a Hamiltonian phase-space coupling: the quadratic known-H case recovers the oracle q-marginal, while the practical unknown-H method uses a learned symplectic predictor and geometry-agnostic non-collapse constraints.

Takeaway. Section 3 does not merely say that Euclidean isotropy can be suboptimal. It shows that, under unknown downstream geometry, fixed marginal matching alone does not specify the predictive problem: no geometry-independent covariance target is universally canonical, and even an oracle marginal does not determine the JEPA predictive coupling. This motivates additional structural assumptions on the view-to-view map.

4 HamJEPA: Symplectic Predictive Structure Without Isotropic Marginals

Section 3 shows that fixed marginal matching is brittle under unknown structured geometry and that marginals alone do not determine JEPA prediction. HamJEPA therefore places the structural bias in the view-to-view coupling: a phase-space encoder is trained with a symplectic, volume-preserving predictor, while encoder-side scale and spectral penalties discourage collapse without enforcing isotropic Gaussian marginals. A visual overview of HamJEPA is provided in Appendix G.

4.1 Phase-space encoder

Given paired views v_a, v_b as defined in Section 2, the encoder $E_\theta : \mathcal{X} \to \mathbb{R}^{2d_0}$ outputs phase-space states $s_a = E_\theta(v_a)$ and $s_b = E_\theta(v_b)$ with $s = (q, p) \in \mathbb{R}^{2d_0}$.

$$s = E_\theta(v) = (q, p), \qquad q, p \in \mathbb{R}^{d_0}. \tag{26}$$

We interpret q as a *content* coordinate used for prediction and evaluation, and p as an auxiliary *momentum* coordinate enabling structured transitions. We write $\pi_q(q, p) = q$ and $\pi_p(q, p) = p$.

Observable coordinate vs. conjugate latent. Section 3 is formulated in terms of the representation used by downstream tasks; in HamJEPA, that conceptual role is played by q. The auxiliary coordinate p is the conjugate latent that makes Hamiltonian transport possible. For non-invertible augmentations, this makes q-matching the cleanest observable formulation, since the target need not contain a uniquely recoverable dynamical state. In practice, however, the exact matching choice is a stability knob rather than a theorem: on CIFAR-100 we primarily use q-matching with an auxiliary $0.2p$ matching term, while on ImageNet-100 we use full-state (q, p)-matching as a pre-evaluation stability choice that avoids an additional matching-mode and momentum-weight search.

4.2 Symplectic predictor via a separable Hamiltonian and leapfrog

Canonical symplectic structure. Let

$$J := \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \in \mathbb{R}^{2d_0 \times 2d_0} \tag{27}$$

denote the canonical symplectic matrix.

Hamilton's equations and flow map. Given a differentiable scalar energy $\mathcal{H}_\phi : \mathbb{R}^{2d_0} \to \mathbb{R}$, Hamilton's ODE on phase space is

$$\dot{s}(t) \;=\; J\,\nabla_s \mathcal{H}_\phi\big(s(t)\big), \qquad s(0) = s_0, \tag{28}$$

where J is the canonical symplectic matrix (27). We denote its time-t flow map, wherever defined, by $\Phi_{\phi,t}$, i.e. $s(t) = \Phi_{\phi,t}(s_0)$.

Separable energy (the theory-tight choice). To obtain a discrete predictor map with rigorous symplectic guarantees under an explicit integrator, we use a separable Hamiltonian energy

$$\mathcal{H}_\phi(q, p) = T(p) + V_\phi(q), \qquad T(p) = \tfrac{1}{2}\|p\|_2^2, \tag{29}$$

where $V_\phi : \mathbb{R}^{d_0} \to \mathbb{R}$ is a learnable potential. Fixing $T(p) = \frac{1}{2}\|p\|^2$ is stabilizing and prevents scale-control constraints from being satisfied by rescaling a learnable kinetic metric.

Leapfrog (velocity Verlet) predictor map. Given step size $\Delta t > 0$ and $K \in \mathbb{N}$ steps, define the leapfrog update

$$\begin{aligned} p_{k+\frac{1}{2}} &= p_k - \tfrac{\Delta t}{2}\,\nabla_q V_\phi(q_k), \\ q_{k+1} &= q_k + \Delta t\, p_{k+\frac{1}{2}}, \qquad\qquad k = 0, \dots, K-1. \\ p_{k+1} &= p_{k+\frac{1}{2}} - \tfrac{\Delta t}{2}\,\nabla_q V_\phi(q_{k+1}), \end{aligned} \tag{30}$$

This defines a one-step map $\Psi_{\phi,\Delta t} : \mathbb{R}^{2d_0} \to \mathbb{R}^{2d_0}$ and a K-step predictor

$$\Phi_\phi^{(K)} := \Psi_{\phi,\Delta t}^{K}. \tag{31}$$

Given $s_a = (q_a, p_a)$, we predict $\hat{s}_b = (\hat{q}_b, \hat{p}_b)$ by

$$\hat{s}_b = \Phi_\phi^{(K)}(s_a). \tag{32}$$

4.3 Objective: content prediction plus non-isotropic anti-collapse

The projected log-det and participation-ratio penalties should be understood as an encoder-side anti-collapse bridge, not as the primary novelty of the method; similar covariance-volume objectives appear elsewhere, whereas our contribution is to couple such non-collapse controls with a phase-space JEPA and a symplectic predictive prior.

Let $s_a = (q_a, p_a) = E_\theta(v_a)$, $s_b = (q_b, p_b) = E_\theta(v_b)$, and $\hat{s}_b = (\hat{q}_b, \hat{p}_b) = \Phi_\phi^{(K)}(s_a)$. HamJEPA optimizes

$$\min_{\theta,\phi} \ \mathcal{L}_{\text{bi}} + \lambda_{\text{budget}}\mathcal{L}_{\text{budget}} + \lambda_{\text{var}}\mathcal{L}_{\text{var}} + \lambda_{\log\det} \sum_{x\in\{q,p\}} \left(\mathcal{L}^x_{\text{vol}} + k^{-2}\mathcal{L}^x_{\text{pr}} + \mathcal{L}^x_{\text{eig}}\right) + \lambda_{\text{mean}}\mathcal{L}_{\text{mean}}. \tag{33}$$

All expectations are implemented as minibatch averages; $\mathrm{sg}(\cdot)$ denotes stop-gradient. MSE averages squared errors over batch and coordinates. The spectral terms are applied to both q and p, with block-specific floors and ceilings; the variance floor is applied to both blocks on CIFAR-100 and to q on ImageNet-100. The weights $(\lambda_{\text{budget}}, \lambda_{\text{var}}, \lambda_{\log\det}, \lambda_{\text{mean}})$ are $(0.2, 0.5, 0.5, 0.05)$ on CIFAR-100 and $(0.2, 1, 1, 0.05)$ on ImageNet-100. Appendix H specifies the variance and mean penalties and all thresholds. Energy consistency is disabled; weight decay is applied by AdamW.

(1) Matching loss. Let $m \in \{q, qp\}$ denote the matching mode, with $\Pi_q(q,p) = q$ and $\Pi_{qp}(q,p) = (q,p)$. CIFAR-100 uses $m = q$ with auxiliary momentum weight $w_p = 0.2$, while ImageNet-100 uses $m = qp$ for stability. Let $\ell_q(s,t) = \mathrm{MSE}(\pi_q(s), \pi_q(t)) + w_p\mathrm{MSE}(\pi_p(s), \pi_p(t))$ and $\ell_{qp}(s,t) = \mathrm{MSE}(s,t)$. We define

$$\mathcal{L}_{\text{pred}} := \mathbb{E}[\ell_m(\hat{s}_b, \mathrm{sg}(s_b))]\,. \tag{34}$$

(2) Bidirectional prediction. Define the backward prediction by running the *same* integrator with step size $-\Delta t$ on CIFAR-100:

$$\hat{s}_a = \Phi_\phi^{(K)}(s_b;\, -\Delta t).$$

The ImageNet-100 implementation instead uses $+\Delta t$ in both view orders. With $\eta = -1$ on CIFAR-100 and $\eta = +1$ on ImageNet-100, the bidirectional loss is

$$\mathcal{L}_{\text{bi}} := \tfrac{1}{2}\mathbb{E}\Big[\ell_m\Big(\Phi_\phi^{(K)}(s_a;\, +\Delta t), \mathrm{sg}(s_b)\Big) + \ell_m\Big(\Phi_\phi^{(K)}(s_b;\, \eta\Delta t), \mathrm{sg}(s_a)\Big)\Big]. \tag{35}$$

Time reversibility of leapfrog (Corollary B.5) makes the CIFAR-100 return map an "exact inverse in-model" constraint rather than an independent predictor. The ImageNet-100 objective averages two forward-time predictions and does not impose that negative-time inverse constraint.

(3) Fixed-units scale budget (anti-gauge). Symplecticity constrains the predictor map but does not prevent the encoder from collapsing all views to a point. We therefore penalize deviations from scale budgets in fixed units (no Gaussianity, no isotropy):

$$\mathcal{L}_{\text{budget}} := \Big(\frac{\mathbb{E}\|q\|_2^2}{d_0} - \alpha_q\Big)^2 + \Big(\frac{\mathbb{E}\|p\|_2^2}{d_0} - \alpha_p\Big)^2, \alpha_q, \alpha_p > 0. \tag{36}$$

The spectral terms apply projected log-det and participation-ratio floors and a top-eigenvalue-fraction ceiling to both q and p. Here k is the projection dimension; Appendix H.8 gives their exact definitions and normalization. Together, these terms discourage collapse while permitting anisotropy; ***HamJEPA*** does not constrain representations to a unit sphere or isotropic Gaussian target.

4.4 Structural guarantees

The formal structural theory of HamJEPA is deferred to Appendix B. At a high level, the separable Hamiltonian (29) and leapfrog update (30) imply that the K-step predictor $\Phi_\phi^{(K)}$ is symplectic:

$$\big(D\Phi_\phi^{(K)}(s)\big)^\top J\big(D\Phi_\phi^{(K)}(s)\big) = J, \qquad \det D\Phi_\phi^{(K)}(s) = 1. \tag{37}$$

Thus the predictor is locally volume-preserving and is inverted by running leapfrog with step size $-\Delta t$. Since symplecticity constrains only the predictor, encoder collapse is controlled separately by the fixed-unit scale, projected log-det, and participation-ratio floors. Appendix B gives reciprocal singular-value pairing, shadow-Hamiltonian stability, and the anti-collapse lemmas.

Setting	Method / readout	kNN@20 (%)	Linear (%)	kNN best (%)
30 epochs, mean±std over 3 seeds				
	SIGReg	26.56 ± 0.18	30.43 ± 0.30	–
	HamJEPA-q	$\mathbf{31.45 \pm 0.27}$	33.95 ± 0.24	–
	HamJEPA-p	29.71 ± 0.25	33.07 ± 0.36	–
	HamJEPA-(q, p)	30.88 ± 0.39	$\mathbf{34.18 \pm 0.17}$	–

Table 1: **CIFAR-100 controlled results.** The 30-epoch block shows short-budget performance. The original single-run 80-epoch results are retained in Table 4.

Method / readout	Dim.	Raw kNN@20	Centered kNN@20	Linear
SIGReg-q	1024	25.61	25.62	32.22
SIGReg-full	2048	27.50	27.53	34.21
MLP-HJEPA-q	1024	27.50	27.48	41.47
VPR-NS-q	1024	34.59	34.33	44.18
Non-reversible VPR-NS-q	1024	34.93	34.89	44.22
HamJEPA-q	1024	34.74	34.59	43.88

Table 2: **CIFAR-100 replicated results, 80 epochs.** Mean accuracy (%) over two training seeds. Each linear result averages five probes per training seed. Sample standard deviations are in Appendix F.4.

5 Results

Controlled protocol. We evaluate frozen representations on CIFAR-100 and ImageNet-100 using cosine kNN and linear probing. The probe uses LayerNorm followed by a linear classifier on frozen features. The goal is not a projector-heavy SOTA recipe, but an objective-level comparison: all methods share the same backbone, augmentations, optimizer, batch size, schedule, two global views, no local views, the encoder runs in a headless token regime using ResNet stage-3 (`layer3`) features, and identity projector. For HamJEPA, the encoder output $s = (q, p)$ is split channel-wise; q is the primary readout, while p and (q, p) test auxiliary-state and dimension effects. Full details and diagnostics are in Appendix F–H. SIGReg is the isotropic-Gaussian JEPA baseline closest to our theoretical comparison. This protocol is intentionally underpowered by modern SSL standards. By removing projection heads, local crops, EMA teachers, and recipe-specific tuning, the comparison evaluates the complete training objectives under a shared minimal protocol.

5.1 CIFAR-100: controlled gains and predictor ablation

HamJEPA improves over SIGReg after only 30 epochs under the same backbone, views, optimizer, and identity projector. Using the primary q readout, HamJEPA gains $+4.89$ points in kNN@20 (31.45 vs. 26.56) and $+3.52$ points in linear probing (33.95 vs. 30.43). The p readout is also predictive (29.71 kNN@20; 33.07 linear), but remains behind q in nearest-neighbor accuracy, matching the intended role of q as the content coordinate and p as an auxiliary dynamical coordinate. Concatenating (q, p) gives the strongest linear probe at 30 epochs (34.18), while q alone gives the strongest kNN geometry.

The original 80-epoch predictor-family ablation gives a single-run comparison, retained in Appendix F.2.

Replicated comparison and matched controls. Table 2 reports two independent 80-epoch training seeds, with five probe optimizations per frozen representation. SIGReg-q is its 1024-dimensional first block; SIGReg-full is the 2048-dimensional baseline used in the original table. HamJEPA retains its gain after centering. A parameter-identical, reversible, volume-preserving non-symplectic force-rotation control (VPR-NS), and its non-reversible second-order splitting control, perform comparably. These results do not identify symplecticity or reversibility as individually necessary. Full metric sweeps, uncertainty, dynamics, sensitivity, and cost are in Appendix F.4.

Method	kNN@20 (%)	Linear probe (%)
SIGReg (q-eval)	20.10	24.40
SIGReg (p-eval)	18.78	23.86
SIGReg ((q, p)-eval)	20.94	25.54
HamJEPA (q-eval)	**24.92**	31.92
HamJEPA (p-eval)	24.72	**32.08**
HamJEPA ((q, p)-eval)	24.64	32.04

Table 3: **ImageNet-100 downstream performance after** 45 **epochs** of pretraining, single seed. For HamJEPA, q and p are learned phase-space coordinates. For SIGReg, the same split is a representation-block / dimension control rather than a literal position–momentum decomposition.

5.2 ImageNet-100: same minimal recipe, larger-scale signal

On ImageNet-100, HamJEPA again improves over SIGReg under the same minimal recipe (45 epochs, single seed). Using the primary q readout, HamJEPA gains $+4.82$ points in kNN@20 (24.92 vs. 20.10) and $+7.52$ points in linear probing (31.92 vs. 24.40). The best HamJEPA linear readout is p at 32.08, giving a $+7.68$ point gain over SIGReg-q. These gains are not explained by feature dimension. SIGReg benefits modestly from concatenating its two 1024-d blocks: $q \to (q, p)$ improves by $+0.84$ kNN@20 points (20.10 $\to$ 20.94) and $+1.14$ linear-probe points (24.40 $\to$ 25.54). In contrast, HamJEPA gains essentially nothing from concatenation in kNN (24.92 $\to$ 24.64) and only $+0.12$ in linear probing (31.92 $\to$ 32.04). Crucially, the 1024-d HamJEPA-q readout already outperforms the 2048-d SIGReg-(q, p) readout by $+3.98$ points in kNN@20 and $+6.38$ points in linear probing.

We match the full phase space (q, p) during pretraining for stability. Consistent with this objective, p is highly predictive and slightly exceeds q on the linear probe (32.08 vs. 31.92), while remaining comparable in kNN (24.72 vs. 24.92). However, concatenating (q, p) does not improve kNN, suggesting that the nearest-neighbor geometry is not simply a consequence of evaluating a larger vector. The result is therefore a controlled objective-level delta: HamJEPA improves frozen-feature geometry in a deliberately headless setting where neither projection-head capacity nor representation dimension explains the gain.

Geometry diagnostics. The accuracy gains are accompanied by broader, non-collapsed covariance spectra without enforcing an isotropic Gaussian marginal. On ImageNet-100, SIGReg-q has effective rank ≈ 38.33 and PR ≈ 24.91, while HamJEPA-q reaches ≈ 94.73 and ≈ 45.27; even the larger SIGReg-(q, p) representation remains lower (≈ 51.16 and ≈ 36.17). CIFAR-100 shows the same qualitative pattern, with HamJEPA-q increasing effective rank from ≈ 109.7 to ≈ 124.5. Full spectra and training-time diagnostics are deferred to Appendix F.

6 Conclusion

We showed that Euclidean-isotropic JEPA regularization can be misaligned with structured task geometry, and that fixed marginal targets are brittle when the downstream geometry is unknown. Under a specified stationary Gaussian lift, reversibility and fixed-unit kinematics characterize the linear Hamiltonian coupling; its quadratic leapfrog discretization has an exact corrected invariant law. HamJEPA instead places Hamiltonian structure in the view-to-view predictor: views are encoded as phase-space states, predicted with a symplectic leapfrog map, and regularized by non-isotropic scale and spectral floors. Under a deliberately headless token protocol, HamJEPA improves frozen-feature kNN and linear-probe performance over SIGReg on CIFAR-100 and ImageNet-100. Matched non-symplectic and non-reversible controls retain the gain, so these experiments do not identify either property as individually necessary.

Limitations. This is a controlled geometry study rather than a SOTA SSL recipe. The main baseline is SIGReg, ImageNet-100 is single-seed, and we do not learn or recover the downstream operator H; broader baselines, multi-seed ImageNet runs, and learned structured geometries remain future work. The new CIFAR-100 controls use two independent training seeds; learned-marginal HamSIGReg and standard projector/EMA/multicrop recipes have not been evaluated.

Acknowledgments and Disclosure of Funding

The author has no funding or competing financial interests to disclose.

References

[1] Shun'ichi Amari. Differential-geometrical methods in statistics, 1985.

[2] 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, 2023.

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

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2021.

[5] Ludwig Boltzmann. Weitere studien über das wärmegleichgewicht unter gasmolekülen, 1872.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[8] J. Willard Gibbs. Elementary principles in statistical mechanics: Developed with especial reference to the rational foundation of thermodynamics, 1902.

[9] Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks, 2019. URL https://arxiv.org/abs/1906.01563.

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[11] William Rowan Hamilton. On a general method in dynamics, 1834.

[12] E. T. Jaynes. Information theory and statistical mechanics, 1957.

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[14] Pierre-Simon Laplace. Traité de mécanique céleste, 1799.

[15] Joseph Liouville. Sur la théorie de la variation des constantes arbitraires, 1838.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[17] Siméon-Denis Poisson. Sur la variation des constantes arbitraires dans les questions de mécanique, 1809.

[18] Peter Toth, Danilo Jimenez Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks, 2019.

[19] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018.

[20] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere, 2020.

[21] Ziyang Wu, Jingyuan Zhang, Druv Pai, XuDong Wang, Chandan Singh, Jianwei Yang, Jianfeng Gao, and Yi Ma. Simplifying dino via coding rate regularization, 2025. URL https://arxiv.org/abs/2502.10385.

[22] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[23] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021.

[24] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control, 2019.

[25] Eric Zimmermann, Harley Wiltzer, Justin Szeto, David Alvarez-Melis, and Lester Mackey. Kerjepa: Kernel discrepancies for euclidean self-supervised learning, 2025. URL https://arxiv.org/abs/2512.19605.

A Societal impact.

HamJEPA is a foundational representation-learning method evaluated on standard image benchmarks, not a deployed decision system. Potential positive impacts include more controlled and sample-efficient visual representation learning. Potential negative impacts are generic to stronger visual representations: downstream systems could be used for surveillance, biometric identification, or biased high-stakes classification if deployed without appropriate safeguards. We recommend dataset and model documentation, license compliance, privacy review, and fairness evaluation before any deployment in sensitive domains.

B Theory: symplecticity, volume preservation, and anti-collapse

This appendix records the structural guarantees enjoyed by HamJEPA when using the separable Hamiltonian (29) and the leapfrog integrator (30). These results justify the summary in Section 4.4: leapfrog gives a symplectic, volume-preserving, time-reversible predictor, while satisfied encoder-side scale, log-det, and participation-ratio constraints give conditional projected-covariance bounds without imposing Euclidean isotropic Gaussian marginals. All proofs are in Appendix C.2.

B.1 Statements

Row/column convention (linear transforms). The paper uses column vectors for single states/embeddings and row-stacked matrices for minibatches. The following lemma formalizes the equivalence.

Lemma B.1 (Row/column convention for left-multiplication by $H^{1/2}$)**.** *Let $H \in \mathbb{R}^{d\times d}$ be symmetric positive definite and let $z \in \mathbb{R}^d$ be a column vector. Define $z' := H^{1/2}z$. Let $\mathbf{Z} \in \mathbb{R}^{n\times d}$ be a row-stacked minibatch with rows $\mathbf{Z}_{i,:} = z_i^\top$. Define $\mathbf{Z}' := \mathbf{Z}H^{1/2}$. Then for every i,*

$$(\mathbf{Z}')_{i,:}^\top \;=\; H^{1/2}z_i.$$

Proof. See Appendix C.2.1. □

Symplecticity of the continuous flow. We first state the standard symplecticity property of Hamiltonian flows.

Theorem B.2 (Symplecticity of Hamiltonian flows)**.** *Assume $\mathcal{H}_\phi \in C^2(\mathbb{R}^{2d_0})$ and let $\Phi_{\phi,t}$ denote the flow map of Hamilton's ODE* (28)*. Then $\Phi_{\phi,t}$ is symplectic: for all s and t for which the flow is defined,*

$$\big(D\Phi_{\phi,t}(s)\big)^\top J\,\big(D\Phi_{\phi,t}(s)\big) \;=\; J. \tag{38}$$

Consequently, $\det(D\Phi_{\phi,t}(s)) = 1$ for all such s, t, and $\Phi_{\phi,t}$ preserves phase-space volume (Liouville).

Proof. See Appendix C.2.2. □

Corollary B.3 (Volume-preserving maps preserve differential entropy)**.** *Let $F : \mathbb{R}^m \to \mathbb{R}^m$ be a C^1 diffeomorphism and let X admit a density. Assume $h(X)$ is finite and $\mathbb{E}|\log|\det DF(X)|| < \infty$. Then*

$$h(F(X)) \;=\; h(X) + \mathbb{E}\big[\log|\det(DF(X))|\big].$$

In particular, if $|\det(DF(x))| = 1$ for all x (e.g., $F = \Phi_{\phi,t}$ from Theorem B.2), then $h(F(X)) = h(X)$.

Proof. See Appendix C.2.3. □

Symplecticity of the discrete predictor. HamJEPA uses the leapfrog/Verlet map (30) as a discrete predictor. For separable Hamiltonians this map is symplectic.

Theorem B.4 (Leapfrog is symplectic and volume-preserving for separable Hamiltonians)**.** *Let $\mathcal{H}_\phi(q,p) = T(p) + V_\phi(q)$ with $T(p) = \frac{1}{2}\|p\|_2^2$ and $V_\phi \in C^2$. Then the one-step leapfrog map $\Psi_{\phi,\Delta t}$ defined by* (30) *is symplectic. Therefore, for any $K \geq 1$, the K-step predictor $\Phi_\phi^{(K)} = \Psi_{\phi,\Delta t}^K$ is symplectic and satisfies*

$$\det\big(D\Phi_\phi^{(K)}(s)\big) = 1 \quad \textit{for all } s. \tag{39}$$

Proof. See Appendix C.2.4. □

Corollary B.5 (Invertibility and time reversibility)**.** *Under the assumptions of Theorem B.4, the leapfrog map is bijective and satisfies*

$$\Psi_{\phi,\Delta t}^{-1} = \Psi_{\phi,-\Delta t}. \tag{40}$$

In particular, the K-step predictor admits an exact inverse by running the same integrator with step size $-\Delta t$: $(\Phi_\phi^{(K)}(\cdot;+\Delta t))^{-1} = \Phi_\phi^{(K)}(\cdot;-\Delta t)$.

Proof. See Appendix C.2.5. □

Proposition B.6 (Approximation of separable Hamiltonian flows by HamJEPA predictors)**.** *Fix a horizon $T > 0$ and a compact set $\mathcal{K} \subset \mathbb{R}^{2d_0}$. Let $\mathcal{K}_q := \{q \in \mathbb{R}^{d_0} : \exists p \text{ with } (q,p) \in \mathcal{K}\}$.*

Let $V : \mathbb{R}^{d_0} \to \mathbb{R}$ be C^3 on an open neighborhood of a compact set $\Omega_q \supseteq \mathcal{K}_q$, and assume ∇V is Lipschitz on Ω_q. Let Φ_t^V denote the time-t flow map of Hamilton's ODE

$$\dot q = p, \qquad \dot p = -\nabla V(q)$$

(i.e., the separable Hamiltonian $H(q,p) = \frac{1}{2}\|p\|_2^2 + V(q)$).

For a learnable potential $V_\phi \in C^3$ with ∇V_ϕ Lipschitz on Ω_q, let $\Psi_{\phi,\Delta t}$ be one leapfrog step (30) *and let $\Phi_\phi^{(K)} := \Psi_{\phi,\Delta t}^K$ with $K\Delta t = T$* (31)*. Assume that for all initial states in $\mathcal{K}$, both the exact trajectory under V and the exact trajectory under V_ϕ remain inside $\Omega := \Omega_q \times \Omega_p$ for all $t \in [0,T]$, where Ω_p is compact. For all sufficiently small $\Delta t = T/K$, assume also that the leapfrog iterates, their split stages, and the exact V_ϕ-flow segments of length at most Δt starting at those iterates remain in Ω.*

Then there exists a constant $C = C(T, \Omega, V_\phi) > 0$ such that

$$\sup_{(q,p)\in\mathcal{K}} \left\|\Phi_\phi^{(K)}(q,p) - \Phi_T^V(q,p)\right\|_2 \;\le$$

$$C\Big(\underbrace{\sup_{q\in\Omega_q} \|\nabla V_\phi(q) - \nabla V(q)\|_2}_{\textit{modeling error}} \;+\; \underbrace{\Delta t^2}_{\textit{integrator error}}\Big).$$

In particular, if the potential class can approximate ∇V uniformly on Ω_q (and admits bounded derivatives on Ω_q, with a common Lipschitz bound for ∇V_ϕ and the stated trajectory containment along the approximating family), then for every $\varepsilon > 0$ there exist ϕ and $\Delta t > 0$ (hence $K = T/\Delta t$) such that

$$\sup_{(q,p)\in\mathcal{K}} \left\|\Phi_\phi^{(K)}(q,p) - \Phi_T^V(q,p)\right\|_2 \le \varepsilon.$$

Proof. See Appendix C.2.6. □

Lemma B.7 (Reciprocal singular-value pairing)**.** *Let $A \in \mathbb{R}^{2d_0\times 2d_0}$ be symplectic: $A^\top J A = J$. Then the singular values of A occur in reciprocal pairs: if σ is a singular value, so is $1/\sigma$ (with the same multiplicity). Equivalently, A cannot be contractive in all directions; any local contraction must be compensated by expansion.*

Proof. See Appendix C.2.7. □

Long-horizon stability (shadow Hamiltonians). A core practical advantage of symplectic integrators is the existence of a nearby modified energy that explains stable multi-step rollouts. We record a standard backward-error-analysis statement.

Theorem B.8 (Shadow Hamiltonian / backward error analysis (standard))**.** *Assume V_ϕ is analytic on a neighborhood of a fixed compact set containing all numerical iterates and modified-flow segments under consideration. For each fixed integer $N \ge 1$ and sufficiently small $\Delta t > 0$, there exists a modified Hamiltonian $\widetilde{\mathcal{H}}_{\phi,\Delta t}^{[N]}$ such that:*

1. ***(Closeness)*** *$\widetilde{\mathcal{H}}_{\phi,\Delta t}^{[N]}(s) = \mathcal{H}_\phi(s) + \mathcal{O}(\Delta t^2)$ uniformly on the region of interest.*

2. ***(Modified-flow representation)*** *One leapfrog step equals the time-Δt flow of $\widetilde{\mathcal{H}}^{[N]}_{\phi,\Delta t}$ up to local error $\mathcal{O}(\Delta t^{N+1})$.*

3. ***(Near-conservation)*** *Along the leapfrog iterates (s_k), with $T_n = n\Delta t$,*

$$|\mathcal{H}_\phi(s_n) - \mathcal{H}_\phi(s_0)| \le C_0 \Delta t^2 + C_1 T_n \Delta t^N,$$

while the stated domain conditions hold. For fixed $N > 2$, this is $\mathcal{O}(\Delta t^2)$ on horizons $T_n = \mathcal{O}(\Delta t^{2-N})$. The constants may depend on the truncation order and controlled region.

Proof (sketch via BCH/backward error analysis). See Appendix C.2.8. □

Anti-collapse: what each ingredient does (and does not do). HamJEPA uses an *encoder-side* non-degeneracy bridge that avoids isotropic Gaussianity. It is useful to separate (i) scale control, (ii) rank/volume control, and (iii) spike control.

Lemma B.9 (Scale budgets do not prevent rank collapse)**.** *Fix $c > 0$ and consider PSD matrices $\Sigma \succeq 0$ with $\mathrm{tr}(\Sigma) = c$. There exist such Σ with rank 1 (hence complete collapse except for one direction), e.g. $\Sigma = \mathrm{diag}(c, 0, \dots, 0)$.*

Proof. See Appendix C.2.9. □

Lemma B.10 (Log-det alone does not prevent "one-spike" spectra)**.** *Fix $k \ge 2$ and $\tau \in \mathbb{R}$. There exist positive definite $\Sigma \in \mathbb{R}^{k\times k}$ with $\frac{1}{k}\log\det(\Sigma) = \tau$ but with $\mathrm{PR}(\Sigma)$ arbitrarily close to 1 (spike regime).*

Proof. Appendix C.2.10. □

Lemma B.11 (Participation ratio alone does not fix scale)**.** *For any PSD $\Sigma \succeq 0$ and any scalar $a > 0$, $\mathrm{PR}(a\Sigma) = \mathrm{PR}(\Sigma)$. Thus a PR constraint alone does not prevent collapse by uniformly shrinking all eigenvalues.*

Proof. See Appendix C.2.11. □

Lemma B.12 (Projected log-det detects full-space rank collapse)**.** *Let $Q \in \mathbb{R}^{d_0}$ be a zero-mean random vector with covariance $\Sigma_Q \succeq 0$ and let $R \in \mathbb{R}^{d_0\times k}$ have full column rank k (e.g., orthonormal columns as in Appendix H.8). Define the population projected covariance*

$$\Sigma_{Y,0} := R^\top \Sigma_Q R \in \mathbb{R}^{k\times k}.$$

Then

$$\mathrm{rank}(\Sigma_{Y,0}) \le \mathrm{rank}(\Sigma_Q).$$

In particular, if $\mathrm{rank}(\Sigma_Q) < k$ then $\Sigma_{Y,0}$ is singular and therefore $\det(\Sigma_{Y,0}) = 0$ and $\log\det(\Sigma_{Y,0}) = -\infty$.

Moreover, if one uses the ridge-regularized version from (47)*,*

$$\Sigma_Y := \Sigma_{Y,0} + \varepsilon I_k, \qquad \varepsilon > 0,$$

and if $r := \mathrm{rank}(\Sigma_{Y,0})$, then Σ_Y has exactly $k - r$ eigenvalues equal to ε and hence

$$\log\det(\Sigma_Y) = (k-r)\log\varepsilon + \sum_{i=1}^{r}\log(\lambda_i + \varepsilon),$$

where $\lambda_1, \dots, \lambda_r > 0$ are the nonzero eigenvalues of $\Sigma_{Y,0}$.

Proof. See Appendix C.2.12. □

Projected log-det floor. Recall the projected covariance Σ_Y from (47). The following proposition quantifies how a log-det floor plus a trace/scale control lower-bounds the smallest eigenvalue.

Proposition B.13 (Projected log-det floor prevents rank collapse)**.** *Let $k \ge 2$ and let $\Sigma_Y \in \mathbb{R}^{k\times k}$ be SPD with eigenvalues $\lambda_1 \ge \cdots \ge \lambda_k > 0$. Suppose $\log\det(\Sigma_Y) \ge k\tau$ and $\mathrm{tr}(\Sigma_Y) \le m$ for some $m > 0$. Then*

$$\lambda_{\min}(\Sigma_Y) \;\ge\; \exp(k\tau)\,\frac{(k-1)^{k-1}}{m^{k-1}}. \tag{41}$$

If $\Sigma_Y = \Sigma_{Y,0} + \varepsilon I_k$, the corresponding unregularized bound is

$$\lambda_{\min}(\Sigma_{Y,0}) \geq \exp(k\tau)\frac{(k-1)^{k-1}}{m^{k-1}} - \varepsilon.$$

It certifies rank k of the projected unregularized covariance only when the right-hand side is positive. When $k < d_0$, this does not certify full ambient rank d_0.

Proof. See Appendix C.2.13. □

Joint spectral non-degeneracy (budget + log-det + PR). The next theorem formalizes the "LogDet + PR + budget" bridge: it rules out both needle collapse and the one-spike "L-shaped" failure mode.

Theorem B.14 (Spectral non-degeneracy via joint constraints)**.** *Let $\Sigma \in \mathbb{R}^{k\times k}$ be SPD with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_k > 0$. Assume the three constraints*

$$\operatorname{tr}(\Sigma) = c, \qquad \operatorname{PR}(\Sigma) \geq r_0, \qquad \frac{1}{k}\log\det(\Sigma) \geq \tau, \tag{42}$$

for some $c > 0$, $r_0 \in (1, k]$, and $\tau \in \mathbb{R}$. Then

$$\lambda_{\max}(\Sigma) \leq \frac{c}{\sqrt{r_0}}; \quad \lambda_{\min}(\Sigma) \geq \exp(k\tau)\left(\frac{\sqrt{r_0}}{c}\right)^{k-1}. \tag{43}$$

Consequently, Σ cannot realize the spike regime $\lambda_1 \gg \lambda_2, \ldots, \lambda_k$ and cannot approach rank collapse $\lambda_{\min} \to 0$ under the stated floors.

Proof. See Appendix C.2.14. □

Corollary B.15 (Bounded conditioning under joint non-degeneracy constraints)**.** *Under the assumptions of Theorem B.14, the condition number satisfies*

$$\kappa(\Sigma) := \frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)} \;\leq\; \exp(-k\tau)\left(\frac{c}{\sqrt{r_0}}\right)^{k}.$$

Proof. See Immediate by dividing the eigenvalue bounds in Theorem B.14. □

Why symplecticity is still meaningful despite encoder regularizers. Symplecticity constrains the predictor map, while the budget/volume/PR constraints constrain the encoder outputs. The next proposition states the correct conditional claim: if the encoder occupies nontrivial volume locally, then the symplectic predictor cannot destroy it.

Proposition B.16 (Symplectic predictors cannot induce collapse)**.** *Fix ϕ and consider the predictor map $\Phi_\phi^{(K)}$ from* (31)*. If the encoder outputs occupy a set $\mathcal{U} \subset \mathbb{R}^{2d_0}$ with positive Lebesgue volume, then $\Phi_\phi^{(K)}(\mathcal{U})$ has the same volume as $\mathcal{U}$. In particular, $\Phi_\phi^{(K)}$ cannot map a positive-volume set to a set of zero volume; any global representational collapse must arise from E_θ, not from the symplectic predictor.*

Proof. See Appendix C.2.15. □

Takeaway. Theorems B.2 and B.4 ensure the predictor map is symplectic and volume-preserving (hence non-contracting in a precise geometric sense), with exact time reversibility (Corollary B.5) and reciprocal singular-value pairing (Lemma B.7). Since symplecticity alone does not prevent encoder collapse, HamJEPA uses a fixed-units scale budget plus projected log-det and PR floors; Theorem B.14 formalizes conditional non-degeneracy bounds when the stated constraints hold. For a ridged projected covariance, the ridge correction in Proposition B.13 is required before inferring an unregularized rank bound. The training penalties encourage these conditions without enforcing isotropic Gaussian marginals.

C Mathematical Proofs

C.1 Proofs for Section 3

C.1.1 Proof of Lemma 3.1

Let $\Sigma \succeq 0$ and $H \succ 0$. For $w \in \mathcal{W}_H$, define $u := H^{-1/2}w$. Then

$$w^\top H^{-1} w = \|H^{-1/2}w\|_2^2 = \|u\|_2^2 \le 1,$$

so $\mathcal{W}_H = \{H^{1/2}u : \ \|u\|_2 \le 1\}$. Substituting $w = H^{1/2}u$ into the objective yields

$$w^\top \Sigma w = u^\top \big(H^{1/2}\Sigma H^{1/2}\big)u.$$

Therefore

$$V(\Sigma) = \sup_{\|u\|_2 \le 1} u^\top \big(H^{1/2}\Sigma H^{1/2}\big)u = \lambda_{\max}\big(H^{1/2}\Sigma H^{1/2}\big),$$

where the last equality is the Rayleigh quotient characterization of the top eigenvalue. If $u_{\max}$ is a unit top eigenvector of $H^{1/2}\Sigma H^{1/2}$, then the corresponding $w_{\max} := H^{1/2}u_{\max}$ satisfies $w_{\max}^\top H^{-1} w_{\max} = 1$ and attains the supremum.

C.1.2 Proof of Theorem 3.2

Let $\Sigma \succeq 0$ and define

$$A := H^{1/2}\Sigma H^{1/2} \succeq 0.$$

Then $\operatorname{tr}(H\Sigma) = \operatorname{tr}(A)$, so the constraint $\operatorname{tr}(H\Sigma) = c$ becomes $\operatorname{tr}(A) = c$. By Lemma 3.1, the objective is $V(\Sigma) = \lambda_{\max}(A)$. Thus (8) is equivalent to

$$\min_{A \succeq 0:\ \operatorname{tr}(A) = c} \lambda_{\max}(A).$$

Let the eigenvalues of A be $\lambda_1 \ge \cdots \ge \lambda_d \ge 0$. Then $\sum_{i=1}^d \lambda_i = c$, hence $\lambda_1 \ge c/d$. Therefore, for every feasible A,

$$\lambda_{\max}(A) \ge \frac{c}{d}.$$

Equality $\lambda_{\max}(A) = c/d$ holds if and only if $\lambda_1 = \cdots = \lambda_d = c/d$, i.e., $A = (c/d)I$, which is unique. Mapping back,

$$\Sigma^\star = H^{-1/2} A H^{-1/2} = \frac{c}{d} H^{-1},$$

and the optimum value is c/d.

C.1.3 Proof of Theorem 3.3

Let Z have a density, $\mathbb{E}[Z] = 0$, and satisfy $\mathbb{E}[Z^\top H Z] = c$. Define the invertible linear change of variables

$$Y := H^{1/2} Z.$$

Then $\mathbb{E}[Y] = 0$ and

$$\mathbb{E}\|Y\|_2^2 = \mathbb{E}[Z^\top H Z] = c.$$

By the change-of-variables formula for differential entropy,

$$h(Z) = h(Y) + \log|\det(H^{-1/2})| = h(Y) - \tfrac{1}{2}\log\det(H).$$

Hence maximizing $h(Z)$ under (11) is equivalent to maximizing $h(Y)$ under $\mathbb{E}\|Y\|_2^2 = c$.

Let $\Sigma_Y := \operatorname{Cov}(Y) \succeq 0$. A standard entropy bound states that for any random vector with covariance Σ_Y,

$$h(Y) \le \frac{1}{2}\log\Big((2\pi e)^d \det(\Sigma_Y)\Big),$$

with equality if and only if Y is Gaussian with covariance Σ_Y.

Next, the constraint $\mathbb{E}\|Y\|_2^2 = c$ implies $\mathrm{tr}(\Sigma_Y) = c$ (since $\mathbb{E}[Y] = 0$). For any PSD matrix with eigenvalues $\nu_1, \ldots, \nu_d \geq 0$ and fixed sum $\sum_i \nu_i = c$, the arithmetic–geometric mean inequality gives

$$\det(\Sigma_Y) = \prod_{i=1}^{d} \nu_i \leq \Big(\frac{1}{d}\sum_{i=1}^{d} \nu_i\Big)^d = \Big(\frac{c}{d}\Big)^d,$$

with equality if and only if $\nu_1 = \cdots = \nu_d = c/d$, i.e., $\Sigma_Y = (c/d)I$.

Combining the two inequalities yields

$$h(Y) \leq \frac{1}{2}\log\Big((2\pi e)^d (c/d)^d\Big),$$

with equality if and only if $Y \sim \mathcal{N}(0, (c/d)I)$. Therefore the unique maximizer is $Y \sim \mathcal{N}(0, (c/d)I)$, and hence

$$Z = H^{-1/2}Y \sim \mathcal{N}\Big(0, \frac{c}{d}H^{-1}\Big),$$

as claimed.

C.1.4 Proof of Theorem 3.4

Under Euclidean isotropy $\Sigma = \sigma^2 I$, the budget $\mathrm{tr}(H\Sigma) = c$ forces

$$\sigma^2 \mathrm{tr}(H) = c \quad \Rightarrow \quad \sigma^2 = \frac{c}{\mathrm{tr}(H)},$$

so $\Sigma_{\text{iso}} = \frac{c}{\mathrm{tr}(H)} I$.

Using Lemma 3.1,

$$V(\Sigma_{\text{iso}}) = \lambda_{\max}\big(H^{1/2}\Sigma_{\text{iso}}H^{1/2}\big) = \lambda_{\max}\Big(\frac{c}{\mathrm{tr}(H)}H\Big) = \frac{c}{\mathrm{tr}(H)}\lambda_{\max}(H),$$

which is (14). From Theorem 3.2, $V(\Sigma^\star) = c/d$, so the ratio is

$$\frac{V(\Sigma_{\text{iso}})}{V(\Sigma^\star)} = \frac{\frac{c}{\mathrm{tr}(H)}\lambda_{\max}(H)}{c/d} = \frac{d\,\lambda_{\max}(H)}{\mathrm{tr}(H)} = \rho(H).$$

Since $\lambda_{\max}(H) \geq \mathrm{tr}(H)/d$ for any SPD H, we have $\rho(H) \geq 1$. Also $\lambda_{\max}(H) \leq \mathrm{tr}(H)$ implies $\rho(H) \leq d$. Finally, $\rho(H) = 1$ holds if and only if $\lambda_{\max}(H) = \mathrm{tr}(H)/d$, which occurs if and only if all eigenvalues of H are equal, i.e. $H \propto I$.

C.1.5 Proof of Proposition 3.5

Let $\mu := \mathbb{E}[Z]$ so that $\Sigma = \mathbb{E}[(Z-\mu)(Z-\mu)^\top]$. Define $Z' := H^{1/2}Z$ and note $\mathbb{E}[Z'] = H^{1/2}\mu$. Then

$$\begin{aligned}
\mathrm{Cov}(Z') &= \mathbb{E}\big[(Z' - \mathbb{E}[Z'])(Z' - \mathbb{E}[Z'])^\top\big] \\
&= \mathbb{E}\big[H^{1/2}(Z-\mu)(Z-\mu)^\top (H^{1/2})^\top\big] \\
&= H^{1/2}\,\mathbb{E}\big[(Z-\mu)(Z-\mu)^\top\big]\,(H^{1/2})^\top \\
&= H^{1/2}\Sigma H^{1/2},
\end{aligned}$$

where we used that $H^{1/2}$ is symmetric for SPD H.

For the equivalence: if $\Sigma = \alpha H^{-1}$ then

$$\mathrm{Cov}(Z') = H^{1/2}(\alpha H^{-1})H^{1/2} = \alpha I.$$

Conversely, if $\mathrm{Cov}(Z') = \alpha I$, multiply left and right by $H^{-1/2}$ to obtain

$$\Sigma = H^{-1/2}(\alpha I)H^{-1/2} = \alpha H^{-1}.$$

□

C.1.6 Proof of Proposition 3.6

By definition,

$$\mathcal{H}_H(q,p) = \tfrac{1}{2}q^\top Hq + \tfrac{1}{2}\|p\|_2^2,$$

so

$$\pi_H(q,p) = \frac{1}{Z_H}\exp\Big(-\frac{d}{2c}q^\top Hq\Big)\exp\Big(-\frac{d}{2c}\|p\|_2^2\Big).$$

Hence the density factorizes into a q-term and a p-term, so q and p are independent.

The partition function also factorizes:

$$Z_H = \Big(\int_{\mathbb{R}^d}\exp\big(-\tfrac{d}{2c}q^\top Hq\big)\,dq\Big)\Big(\int_{\mathbb{R}^d}\exp\big(-\tfrac{d}{2c}\|p\|_2^2\big)\,dp\Big).$$

Using the standard Gaussian integral

$$\int_{\mathbb{R}^d}\exp\big(-\tfrac{1}{2}x^\top Ax\big)\,dx = (2\pi)^{d/2}\det(A)^{-1/2} \qquad (A\succ 0),$$

with $A = (d/c)H$ for the q-integral and $A = (d/c)I$ for the p-integral, we obtain

$$Z_H = \Big(2\pi\frac{c}{d}\Big)^d\det(H)^{-1/2}.$$

Therefore

$$\pi_H(q,p) = \underbrace{\Big(2\pi\frac{c}{d}\Big)^{-d/2}\det(H)^{1/2}\exp\Big(-\frac{d}{2c}q^\top Hq\Big)}_{q\text{-marginal}}\cdot\underbrace{\Big(2\pi\frac{c}{d}\Big)^{-d/2}\exp\Big(-\frac{d}{2c}\|p\|_2^2\Big)}_{p\text{-marginal}}.$$

Thus q is Gaussian with precision matrix $(d/c)H$, hence covariance $(c/d)H^{-1}$, and p is Gaussian with precision $(d/c)I$, hence covariance $(c/d)I$:

$$q\sim\mathcal{N}\Big(0,\frac{c}{d}H^{-1}\Big), \qquad p\sim\mathcal{N}\Big(0,\frac{c}{d}I\Big).$$

Next,

$$\mathbb{E}_{\pi_H}[q^\top Hq] = \operatorname{tr}\Big(H\operatorname{Cov}(q)\Big) = \operatorname{tr}\Big(H\,\frac{c}{d}H^{-1}\Big) = \frac{c}{d}\operatorname{tr}(I) = c.$$

Similarly,

$$\mathbb{E}_{\pi_H}[\|p\|_2^2] = \operatorname{tr}\big(\operatorname{Cov}(p)\big) = \operatorname{tr}\Big(\frac{c}{d}I\Big) = c.$$

Hence

$$\mathbb{E}_{\pi_H}[\mathcal{H}_H(q,p)] = \frac{1}{2}\,\mathbb{E}_{\pi_H}[q^\top Hq] + \frac{1}{2}\,\mathbb{E}_{\pi_H}[\|p\|_2^2] = \frac{1}{2}c + \frac{1}{2}c = c.$$

Finally, the q-marginal is exactly the Gaussian

$$\mathcal{N}\Big(0,\frac{c}{d}H^{-1}\Big),$$

which is the minimax-optimal covariance from Theorem 3.2 and the maximum-entropy law from Theorem 3.3. □

C.1.7 Proof of Proposition 3.7

Let $\Sigma\succ 0$ be arbitrary and define

$$H := \frac{c}{d}\Sigma^{-1}.$$

Since $\Sigma^{-1}\succ 0$, we have $H\succ 0$. Then

$$\Sigma^\star(H) = \frac{c}{d}H^{-1} = \frac{c}{d}\Big(\frac{c}{d}\Sigma^{-1}\Big)^{-1} = \frac{c}{d}\cdot\frac{d}{c}\Sigma = \Sigma.$$

Thus every $\Sigma\succ 0$ arises as $\Sigma^\star(H)$ for some $H\succ 0$, so the family $\{\Sigma^\star(H) : H\succ 0\}$ is exactly the SPD cone. □

C.1.8 Proof of Theorem 3.8

First note that $\Sigma_M(H)$ is feasible for the H-budget:

$$\operatorname{tr}(H\Sigma_M(H)) = \operatorname{tr}\left(H\,\frac{c}{\operatorname{tr}(HM)}M\right) = c.$$

By the spectral form of the worst-case task variance,

$$V_H(\Sigma) = \lambda_{\max}\left(H^{1/2}\Sigma H^{1/2}\right).$$

Therefore

$$V_H(\Sigma_M(H)) = \lambda_{\max}\left(H^{1/2}\frac{c}{\operatorname{tr}(HM)}MH^{1/2}\right) = \frac{c}{\operatorname{tr}(HM)}\lambda_{\max}\left(H^{1/2}MH^{1/2}\right).$$

For the oracle covariance

$$\Sigma^\star(H) = \frac{c}{d}H^{-1},$$

we have

$$H^{1/2}\Sigma^\star(H)H^{1/2} = \frac{c}{d}I,$$

so

$$V_H(\Sigma^\star(H)) = \frac{c}{d}.$$

Dividing the two expressions gives

$$\frac{V_H(\Sigma_M(H))}{V_H(\Sigma^\star(H))} = \frac{\frac{c}{\operatorname{tr}(HM)}\lambda_{\max}(H^{1/2}MH^{1/2})}{c/d} = \frac{d\,\lambda_{\max}(H^{1/2}MH^{1/2})}{\operatorname{tr}(HM)}.$$

Since

$$\operatorname{tr}(HM) = \operatorname{tr}\left(H^{1/2}MH^{1/2}\right),$$

writing

$$A := H^{1/2}MH^{1/2} \succ 0$$

gives

$$\frac{V_H(\Sigma_M(H))}{V_H(\Sigma^\star(H))} = \frac{d\,\lambda_{\max}(A)}{\operatorname{tr}(A)}.$$

For any $A \succ 0$,

$$\lambda_{\max}(A) \le \operatorname{tr}(A) \le d\,\lambda_{\max}(A),$$

and hence

$$1 \le \frac{d\,\lambda_{\max}(A)}{\operatorname{tr}(A)} \le d.$$

The ratio equals 1 if and only if

$$\operatorname{tr}(A) = d\,\lambda_{\max}(A),$$

which holds if and only if all eigenvalues of A are equal, i.e.

$$A = \alpha I$$

for some $\alpha > 0$. Since $A = H^{1/2}MH^{1/2}$, this is equivalent to

$$H^{1/2}MH^{1/2} = \alpha I,$$

or, multiplying on the left and right by $H^{-1/2}$,

$$M = \alpha H^{-1}.$$

Thus the regret is 1 if and only if $M \propto H^{-1}$.

It remains to show that, for every fixed $M \succ 0$, the supremum over $H \succ 0$ is d. The upper bound d was just shown. To see that it is sharp, fix $\delta \in (0,1)$ and define

$$A_\delta := \operatorname{diag}(1, \delta, \ldots, \delta) \succ 0.$$

We claim there exists $H_\delta \succ 0$ such that

$$H_\delta^{1/2} M H_\delta^{1/2} = A_\delta.$$

Indeed, let

$$B_\delta := M^{-1/2} \left(M^{1/2} A_\delta M^{1/2} \right)^{1/2} M^{-1/2} \succ 0$$

and set

$$H_\delta := B_\delta^2.$$

Then $H_\delta^{1/2} = B_\delta$ and

$$H_\delta^{1/2} M H_\delta^{1/2} = B_\delta M B_\delta = A_\delta.$$

For this choice,

$$\frac{V_{H_\delta}(\Sigma_M(H_\delta))}{V_{H_\delta}(\Sigma^\star(H_\delta))} = \frac{d\,\lambda_{\max}(A_\delta)}{\mathrm{tr}(A_\delta)} = \frac{d}{1+(d-1)\delta}.$$

Letting $\delta \downarrow 0$ gives

$$\sup_{H \succ 0} \frac{V_H(\Sigma_M(H))}{V_H(\Sigma^\star(H))} = d.$$

For $d = 1$, the ratio is identically $1 = d$, so every target shape is optimal; the no-universal-shape conclusion requires $d \geq 2$. Thus no geometry-independent fixed covariance shape M is canonical for all structured geometries.

C.1.9 Proof of Proposition 3.9

Because (Z_a, Z_b) is jointly Gaussian, any affine transformation of (Z_a, Z_b) is also Gaussian. Define the residual

$$R := Z_b - C\Sigma^{-1} Z_a.$$

Then

$$\mathrm{Cov}(R, Z_a) = \mathrm{Cov}(Z_b, Z_a) - C\Sigma^{-1}\mathrm{Cov}(Z_a, Z_a) = C - C\Sigma^{-1}\Sigma = 0.$$

Since (R, Z_a) is jointly Gaussian and uncorrelated, R and Z_a are independent. Also, because the variables are centered,

$$\mathbb{E}[R] = 0.$$

Therefore

$$\mathbb{E}[Z_b \mid Z_a] = \mathbb{E}[C\Sigma^{-1} Z_a + R \mid Z_a] = C\Sigma^{-1} Z_a + \mathbb{E}[R \mid Z_a] = C\Sigma^{-1} Z_a.$$

It remains to show that the same marginal covariance Σ admits many different predictive maps. For any $t \in (-1, 1)$, consider the block covariance

$$\Gamma_t := \begin{pmatrix} \Sigma & t\Sigma \\ t\Sigma & \Sigma \end{pmatrix}.$$

For any $u, v \in \mathbb{R}^d$,

$$\begin{pmatrix} u \\ v \end{pmatrix}^\top \Gamma_t \begin{pmatrix} u \\ v \end{pmatrix} = u^\top \Sigma u + 2t\, u^\top \Sigma v + v^\top \Sigma v.$$

Equivalently,

$$u^\top \Sigma u + 2t\, u^\top \Sigma v + v^\top \Sigma v = \frac{1+t}{2}(u+v)^\top \Sigma (u+v) + \frac{1-t}{2}(u-v)^\top \Sigma (u-v).$$

Since $\Sigma \succ 0$ and $t \in (-1, 1)$, this quadratic form is strictly positive for every nonzero (u, v). Hence $\Gamma_t \succ 0$, so it defines a valid centered joint Gaussian law with both marginals equal to Σ and cross-covariance

$$C_t = t\Sigma.$$

For this joint law, the Bayes predictor is

$$\mathbb{E}[Z_b \mid Z_a] = C_t \Sigma^{-1} Z_a = t Z_a.$$

As t varies over $(-1, 1)$, the marginal covariance of both views remains exactly Σ, but the predictive map varies continuously from nearly $-Z_a$ to nearly Z_a. Thus even an oracle marginal covariance does not identify the view-to-view coupling. Marginal matching alone therefore cannot determine the JEPA predictive problem.

C.2 Proofs for Section 4.4

C.2.1 Proof of Lemma B.1

By definition, $\mathbf{Z}' = \mathbf{Z}H^{1/2}$, so its ith row is

$$(\mathbf{Z}')_{i,:} \;=\; (\mathbf{Z}_{i,:})H^{1/2} \;=\; z_i^\top H^{1/2}.$$

Taking transpose gives

$$(\mathbf{Z}')_{i,:}^\top \;=\; (z_i^\top H^{1/2})^\top \;=\; (H^{1/2})^\top z_i \;=\; H^{1/2} z_i,$$

where we used symmetry of $H^{1/2}$ (valid for SPD H).

C.2.2 Proof of Theorem B.2

Fix $s_0 \in \mathbb{R}^{2d_0}$ and let $s(t) = \Phi_{\phi,t}(s_0)$ solve $\dot{s} = J\nabla\mathcal{H}_\phi(s)$. Let $A(t) := D\Phi_{\phi,t}(s_0) \in \mathbb{R}^{2d_0 \times 2d_0}$ denote the Jacobian of the flow with respect to the initial condition. Differentiating the ODE with respect to s_0 yields the variational equation

$$\dot{A}(t) \;=\; J\,\nabla^2\mathcal{H}_\phi(s(t))\,A(t), \qquad A(0) = I. \tag{44}$$

Define $M(t) := A(t)^\top JA(t)$. Using (44) and the facts $J^\top = -J$ and $\nabla^2\mathcal{H}_\phi$ is symmetric, we compute

$$\begin{aligned}\dot{M}(t) &= \dot{A}(t)^\top JA(t) + A(t)^\top J\dot{A}(t)\\ &= A(t)^\top\nabla^2\mathcal{H}_\phi(s(t))^\top J^\top JA(t) + A(t)^\top JJ\nabla^2\mathcal{H}_\phi(s(t))A(t).\end{aligned}$$

Since $\nabla^2\mathcal{H}_\phi(s(t))^\top = \nabla^2\mathcal{H}_\phi(s(t))$ and $J^\top = -J$ and $JJ = -I$, this becomes

$$\begin{aligned}\dot{M}(t) &= A(t)^\top\nabla^2\mathcal{H}_\phi(s(t))\,(-J)JA(t) + A(t)^\top(-I)\nabla^2\mathcal{H}_\phi(s(t))A(t)\\ &= A(t)^\top\nabla^2\mathcal{H}_\phi(s(t))\,A(t) - A(t)^\top\nabla^2\mathcal{H}_\phi(s(t))\,A(t)\\ &= 0.\end{aligned}$$

Thus $M(t)$ is constant in t. At $t = 0$, $A(0) = I$ so $M(0) = J$, hence $M(t) = J$ for all t, proving (38).

For the determinant claim, take determinants of $A(t)^\top JA(t) = J$:

$$\det(A(t))^2\det(J) = \det(J) \quad\Rightarrow\quad \det(A(t))^2 = 1.$$

Since $A(0) = I$ has $\det(A(0)) = 1$ and $\det(A(t))$ varies continuously with t, we must have $\det(A(t)) = 1$ for all t.

Finally, volume preservation follows from $\det(D\Phi_{\phi,t}(s)) \equiv 1$ by the change-of-variables theorem: for any measurable set U in the domain of $\Phi_{\phi,t}$,

$$\mathrm{Vol}(\Phi_{\phi,t}(U)) = \int_U |\det(D\Phi_{\phi,t}(s))|\,ds = \int_U 1\,ds = \mathrm{Vol}(U).$$

C.2.3 Proof of Corollary B.3

Let $Y = F(X)$ with F a C^1 diffeomorphism. The change-of-variables formula gives $p_Y(y) = p_X(x)\,|\det(DF(x))|^{-1}$ with $x = F^{-1}(y)$. Then

$$\begin{aligned}h(Y) &= -\int p_Y(y)\log p_Y(y)\,dy\\ &= -\int p_X(x)\log\left(p_X(x)\,|\det(DF(x))|^{-1}\right)dx\\ &= h(X) + \mathbb{E}[\log|\det(DF(X))|].\end{aligned}$$

If $|\det(DF(\cdot))| \equiv 1$, the second term is zero, so $h(Y) = h(X)$.

C.2.4 Proof of Theorem B.4

Write the leapfrog step as a composition of two "kicks" and one "drift". Define for $h \in \mathbb{R}$:

$$K_h(q,p) := (q,\ p - h\nabla V_\phi(q)), \qquad D_h(q,p) := (q + hp,\ p),$$

where the drift uses $\nabla T(p) = p$ for $T(p) = \frac{1}{2}\|p\|^2$. Then the one-step leapfrog map satisfies

$$\Psi_{\phi,\Delta t} \ = \ K_{\Delta t/2} \circ D_{\Delta t} \circ K_{\Delta t/2}.$$

We show that both K_h and D_h are symplectic. Their Jacobians have block forms

$$DK_h(q,p) = \begin{pmatrix} I & 0 \\ -h\nabla^2 V_\phi(q) & I \end{pmatrix} = \begin{pmatrix} I & 0 \\ B & I \end{pmatrix}, \quad B^\top = B,$$
$$DD_h(q,p) = \begin{pmatrix} I & hI \\ 0 & I \end{pmatrix} = \begin{pmatrix} I & C \\ 0 & I \end{pmatrix}, \quad C^\top = C.$$

A direct calculation shows that any matrix of the form $\left(\begin{smallmatrix} I & 0 \\ B & I \end{smallmatrix}\right)$ with B symmetric satisfies $A^\top J A = J$, and similarly for $\left(\begin{smallmatrix} I & C \\ 0 & I \end{smallmatrix}\right)$ with C symmetric. Therefore DK_h and DD_h are symplectic matrices at every point, so K_h and D_h are symplectic maps.

The composition of symplectic maps is symplectic, hence $\Psi_{\phi,\Delta t}$ is symplectic, and iterating yields symplecticity of $\Phi_\phi^{(K)} = \Psi_{\phi,\Delta t}^{K}$. The determinant claim follows as in the proof of Theorem B.2.

C.2.5 Proof of Corollary B.5

Using the decomposition $\Psi_{\Delta t} = K_{\Delta t/2} \circ D_{\Delta t} \circ K_{\Delta t/2}$, we have $K_h^{-1} = K_{-h}$ and $D_h^{-1} = D_{-h}$ by inspection. Hence

$$\Psi_{\Delta t}^{-1} = K_{\Delta t/2}^{-1} \circ D_{\Delta t}^{-1} \circ K_{\Delta t/2}^{-1} = K_{-\Delta t/2} \circ D_{-\Delta t} \circ K_{-\Delta t/2} = \Psi_{-\Delta t}.$$

Taking K-fold compositions gives $(\Psi_{\Delta t}^{K})^{-1} = \Psi_{-\Delta t}^{K}$.

C.2.6 Proof of Proposition B.6

Write the phase-space state as $s = (q,p) \in \mathbb{R}^{2d_0}$ and define the two vector fields

$$f(s) := \begin{pmatrix} p \\ -\nabla V(q) \end{pmatrix}, \qquad f_\phi(s) := \begin{pmatrix} p \\ -\nabla V_\phi(q) \end{pmatrix}.$$

Let Φ_t^V denote the flow of $\dot{s} = f(s)$ and $\Phi_t^{V_\phi}$ the flow of $\dot{s} = f_\phi(s)$.

Throughout, work on the compact set $\Omega = \Omega_q \times \Omega_p$ from the proposition statement. Since ∇V and ∇V_ϕ are Lipschitz on Ω_q and Ω_p is compact, both f and f_ϕ are Lipschitz on Ω; let L be any Lipschitz constant of f_ϕ on Ω:

$$\|f_\phi(x) - f_\phi(y)\|_2 \le L\|x - y\|_2, \qquad \forall x, y \in \Omega.$$

Define the uniform force mismatch on Ω_q:

$$\delta := \sup_{q \in \Omega_q} \|\nabla V_\phi(q) - \nabla V(q)\|_2.$$

Step 1: Continuous dependence of flows on the force field. Fix $s_0 \in \mathcal{K}$ and let $s(t) := \Phi_t^V(s_0)$ and $\tilde{s}(t) := \Phi_t^{V_\phi}(s_0)$. By assumption, both trajectories remain in Ω for $t \in [0,T]$. Let $\Delta(t) := \tilde{s}(t) - s(t)$. Then

$$\begin{aligned} \dot{\Delta}(t) &= f_\phi(\tilde{s}(t)) - f(s(t)) \\ &= \big(f_\phi(\tilde{s}(t)) - f_\phi(s(t))\big) + \big(f_\phi(s(t)) - f(s(t))\big). \end{aligned}$$

Taking norms and using Lipschitzness of f_ϕ on Ω plus the definition of δ gives

$$\|\dot{\Delta}(t)\|_2 \le L\|\Delta(t)\|_2 + \delta.$$

Since $\Delta(0) = 0$, Grönwall's inequality yields, for all $t \in [0, T]$,

$$\|\Delta(t)\|_2 \le \begin{cases} \dfrac{\delta}{L}\big(e^{Lt} - 1\big), & L > 0, \\ \delta t, & L = 0. \end{cases}$$

In particular, at $t = T$ we obtain a uniform (over $s_0 \in \mathcal{K}$) modeling-error bound

$$\sup_{s_0 \in \mathcal{K}} \|\Phi_T^{V_\phi}(s_0) - \Phi_T^{V}(s_0)\|_2 \le C_1\, \delta, \qquad C_1 := \begin{cases} \frac{e^{LT}-1}{L}, & L > 0, \\ T, & L = 0. \end{cases}$$

Step 2: Leapfrog global error is $O(\Delta t^2)$. Fix ϕ and consider the ODE $\dot{s} = f_\phi(s)$. Let $\Psi_{\phi,h}$ denote one leapfrog step with step size h (Eq. (30)), and define the numerical iterates

$$s_{n+1} := \Psi_{\phi,h}(s_n), \qquad s_0 \in \mathcal{K}, \qquad t_n := nh,$$

so that $s_K = (\Psi_{\phi,h})^K(s_0) = \Phi_\phi^{(K)}(s_0)$ with $Kh = T$.

(2a) Local truncation error bound. Because $V_\phi \in C^3$ on a neighborhood of Ω_q, the vector fields

$$A(q,p) := \begin{pmatrix} p \\ 0 \end{pmatrix}, \qquad B(q,p) := \begin{pmatrix} 0 \\ -\nabla V_\phi(q) \end{pmatrix}$$

are C^2 on Ω, and their exact flows are

$$\exp(hA) : (q,p) \mapsto (q + hp, p), \qquad \exp(hB) : (q,p) \mapsto (q, p - h\nabla V_\phi(q)).$$

The leapfrog step is exactly the Strang splitting:

$$\Psi_{\phi,h} = \exp\left(\frac{h}{2}B\right) \exp(hA) \exp\left(\frac{h}{2}B\right).$$

Standard Strang splitting theory (derivable via the BCH expansion for the Lie operators L_A, L_B) implies that the local truncation error is third order: there exists a constant $C_{\mathrm{loc}} = C_{\mathrm{loc}}(\Omega, V_\phi)$ such that

$$\sup_{x \in \Omega} \big\|\Psi_{\phi,h}(x) - \Phi_h^{V_\phi}(x)\big\|_2 \le C_{\mathrm{loc}}\, h^3 \quad \text{for all sufficiently small } h.$$

(Concretely, the BCH expansion gives $\exp(\frac{h}{2}L_B)\exp(hL_A)\exp(\frac{h}{2}L_B) = \exp\big(h(L_A + L_B) + O(h^3)\big)$ as operators on smooth test functions, and boundedness of the commutators on Ω yields the uniform C_{loc}.)

(2b) Error recursion and discrete Grönwall. Define the global error at step n by

$$e_n := s_n - \Phi_{t_n}^{V_\phi}(s_0).$$

Then

$$\begin{aligned} e_{n+1} &= \Psi_{\phi,h}(s_n) - \Phi_{t_{n+1}}^{V_\phi}(s_0) \\ &= \Psi_{\phi,h}(s_n) - \Phi_h^{V_\phi}\big(\Phi_{t_n}^{V_\phi}(s_0)\big). \end{aligned}$$

Add and subtract $\Phi_h^{V_\phi}(s_n)$:

$$e_{n+1} = \underbrace{\big(\Psi_{\phi,h}(s_n) - \Phi_h^{V_\phi}(s_n)\big)}_{\text{local truncation}} + \underbrace{\big(\Phi_h^{V_\phi}(s_n) - \Phi_h^{V_\phi}(\Phi_{t_n}^{V_\phi}(s_0))\big)}_{\text{stability}}.$$

Taking norms, using the local error bound from (2a), and using Lipschitz stability of the flow $\|\Phi_h^{V_\phi}(x) - \Phi_h^{V_\phi}(y)\|_2 \le e^{Lh}\|x - y\|_2$ on Ω yields

$$\|e_{n+1}\|_2 \le C_{\mathrm{loc}} h^3 + e^{Lh}\|e_n\|_2.$$

Iterating this recursion with $e_0 = 0$ gives

$$\|e_K\|_2 \le C_{\mathrm{loc}} h^3 \sum_{j=0}^{K-1} e^{Lhj} = C_{\mathrm{loc}} h^3 \cdot \begin{cases} \dfrac{e^{LKh} - 1}{e^{Lh} - 1}, & L > 0, \\ K, & L = 0. \end{cases}$$

Since $Kh = T$ and $e^{Lh} - 1 \geq Lh$ for $Lh \geq 0$, we obtain

$$\|e_K\|_2 \leq \begin{cases} \dfrac{C_{\mathrm{loc}}}{L}\left(e^{LT} - 1\right) h^2, & L > 0, \\ C_{\mathrm{loc}} T\, h^2, & L = 0. \end{cases}$$

Taking the supremum over $s_0 \in \mathcal{K}$ (all iterates stay in Ω by assumption), we conclude that there exists $C_2 = C_2(T, \Omega, V_\phi)$ such that

$$\sup_{s_0 \in \mathcal{K}} \left\| \Phi_\phi^{(K)}(s_0) - \Phi_T^{V_\phi}(s_0) \right\|_2 \leq C_2\, h^2.$$

Step 3: Combine modeling and integrator errors. By the triangle inequality,

$$\|\Phi_\phi^{(K)}(s_0) - \Phi_T^V(s_0)\|_2 \leq \|\Phi_\phi^{(K)}(s_0) - \Phi_T^{V_\phi}(s_0)\|_2 + \|\Phi_T^{V_\phi}(s_0) - \Phi_T^V(s_0)\|_2.$$

Taking suprema over $s_0 \in \mathcal{K}$ and applying the bounds from Steps 1–2 yields

$$\sup_{s_0 \in \mathcal{K}} \|\Phi_\phi^{(K)}(s_0) - \Phi_T^V(s_0)\|_2 \leq C_2 h^2 + C_1 \delta.$$

This is the claimed inequality with $C := \max\{C_1, C_2\}$ (or $C := C_1 + C_2$).

Finally, if the model class can make δ arbitrarily small (uniform approximation of ∇V on Ω_q), the assumed common Lipschitz bound keeps C_1 uniformly bounded along the approximating family. Choose ϕ so that $C_1 \delta \leq \varepsilon/2$, then choose $h = \Delta t$ so that $C_2 h^2 \leq \varepsilon/2$, and conclude

$$\sup_{s_0 \in \mathcal{K}} \|\Phi_\phi^{(K)}(s_0) - \Phi_T^V(s_0)\|_2 \leq \varepsilon.$$

□

C.2.7 Proof of Lemma B.7

Let A be symplectic: $A^\top J A = J$. Using $J^{-1} = -J$ and $J^\top = -J$, we obtain

$$A^{-1} = -J A^\top J.$$

Then

$$(A^\top A)^{-1} = A^{-1} A^{-T} = (-J A^\top J)(-JAJ) = J^{-1}(A^\top A) J,$$

so $(A^\top A)^{-1}$ is similar to $A^\top A$ and thus has the same eigenvalues. Therefore the eigenvalues of $A^\top A$ occur in reciprocal pairs $(\mu, 1/\mu)$, and the singular values $\sigma = \sqrt{\mu}$ occur in reciprocal pairs $(\sigma, 1/\sigma)$.

C.2.8 Proof sketch of Theorem B.8

This is a standard result from backward error analysis for symplectic integrators; we provide the BCH-based sketch specialized to leapfrog.

Let L_T and L_V denote the Lie derivatives (Poisson bracket operators) corresponding to the Hamiltonians T and V, acting on smooth observables. The exact time-h flow of T (resp. V) is $\exp(hL_T)$ (resp. $\exp(hL_V)$). The leapfrog step can be written operator-theoretically as the symmetric composition

$$\exp\left(\tfrac{h}{2} L_V\right) \exp(hL_T) \exp\left(\tfrac{h}{2} L_V\right).$$

By the Baker–Campbell–Hausdorff (BCH) formula, this equals

$$\exp\Big(h(L_T + L_V) + h^3 C_3 + h^5 C_5 + \cdots\Big),$$

where $C_3, C_5, \ldots$ are nested commutators of L_T and L_V. For Hamiltonian systems, commutators of Hamiltonian vector fields correspond to Hamiltonian vector fields of Poisson brackets; equivalently, each C_{2j+1} is itself a Lie derivative $L_{H_{2j}}$ for some scalar function H_{2j} built from iterated Poisson brackets of T and V. Thus the exponential above is the exact time-h flow of a *modified Hamiltonian*

$$\widetilde{\mathcal{H}}_{\Delta t} \;=\; \mathcal{H} + h^2 H_2 + h^4 H_4 + \cdots$$

at the level of formal series. Truncating this series at order N yields a modified Hamiltonian $\widetilde{\mathcal{H}}_{\Delta t}^{[N]}$ whose flow matches one leapfrog step up to local error $\mathcal{O}(h^{N+1})$. Under analyticity, classical results show that the truncation can be chosen so that the resulting modified energy is nearly conserved over exponentially long times in $1/h$. (See, e.g., standard references on geometric numerical integration for full details.)

For the stated finite-order bound, fix N. The truncated modified energy has uniformly bounded gradient on the controlled neighborhood for sufficiently small h. Its exact modified flow preserves that energy, whereas the numerical step differs from the modified flow by $\mathcal{O}(h^{N+1})$. Hence

$$|\widetilde{\mathcal{H}}^{[N]}(s_{j+1}) - \widetilde{\mathcal{H}}^{[N]}(s_j)| \le C_1 h^{N+1}.$$

Summing over n steps and using the $\mathcal{O}(h^2)$ difference between the original and modified energies at both endpoints gives

$$|\mathcal{H}_\phi(s_n) - \mathcal{H}_\phi(s_0)| \le C_0 h^2 + C_1 n h^{N+1} = C_0 h^2 + C_1 T_n h^N.$$

C.2.9 Proof of Lemma B.9

Take $\Sigma = \mathrm{diag}(c, 0, \ldots, 0)$. Then $\Sigma \succeq 0$, $\mathrm{tr}(\Sigma) = c$, and $\mathrm{rank}(\Sigma) = 1$.

C.2.10 Proof of Lemma B.10

Fix $k \ge 2$ and τ. For $\varepsilon > 0$, define eigenvalues

$$\lambda_2 = \cdots = \lambda_k = \varepsilon, \qquad \lambda_1 = \exp(k\tau)\, \varepsilon^{-(k-1)}.$$

Then $\prod_{i=1}^k \lambda_i = \exp(k\tau)$, hence $\frac{1}{k} \log\det(\Sigma) = \tau$. But as $\varepsilon \to 0$, $\lambda_1 \to \infty$ and $\lambda_2, \ldots, \lambda_k \to 0$. The participation ratio satisfies

$$\mathrm{PR}(\Sigma) = \frac{(\sum_i \lambda_i)^2}{\sum_i \lambda_i^2} \longrightarrow \frac{\lambda_1^2}{\lambda_1^2} = 1,$$

so $\mathrm{PR}(\Sigma)$ can be made arbitrarily close to 1 while maintaining the log-det constraint.

C.2.11 Proof of Lemma B.11

Let λ_i be the eigenvalues of Σ. Then the eigenvalues of $a\Sigma$ are $a\lambda_i$, so

$$\mathrm{PR}(a\Sigma) = \frac{(\sum_i a\lambda_i)^2}{\sum_i (a\lambda_i)^2} = \frac{a^2 (\sum_i \lambda_i)^2}{a^2 \sum_i \lambda_i^2} = \mathrm{PR}(\Sigma).$$

C.2.12 Proof of Lemma B.12

Let $\Sigma_Q \succeq 0$ and $R \in \mathbb{R}^{d_0 \times k}$ have full column rank. Write $\Sigma_Q = A^\top A$ with $A := \Sigma_Q^{1/2}$ (PSD square root). Then

$$\Sigma_{Y,0} = R^\top \Sigma_Q R = R^\top A^\top A R = (AR)^\top (AR).$$

Therefore $\Sigma_{Y,0}$ is PSD and

$$\mathrm{rank}(\Sigma_{Y,0}) = \mathrm{rank}\left((AR)^\top (AR)\right) = \mathrm{rank}(AR) \le \mathrm{rank}(A) = \mathrm{rank}(\Sigma_Q).$$

If $\mathrm{rank}(\Sigma_Q) < k$, then $\mathrm{rank}(\Sigma_{Y,0}) < k$, so $\Sigma_{Y,0}$ is singular. Hence $\det(\Sigma_{Y,0}) = 0$ and $\log\det(\Sigma_{Y,0}) = -\infty$.

For the ridge statement, let $r := \mathrm{rank}(\Sigma_{Y,0})$ and let $\lambda_1, \ldots, \lambda_r > 0$ be its nonzero eigenvalues. Then $\Sigma_{Y,0}$ has exactly $k - r$ zero eigenvalues. Adding εI_k shifts each eigenvalue by ε, so $\Sigma_Y = \Sigma_{Y,0} + \varepsilon I_k$ has eigenvalues

$$\lambda_1 + \varepsilon,\ \ldots,\ \lambda_r + \varepsilon,\ \underbrace{\varepsilon, \ldots, \varepsilon}_{k-r \text{ times}}.$$

Thus

$$\log\det(\Sigma_Y) = \sum_{i=1}^r \log(\lambda_i + \varepsilon) + (k - r)\log\varepsilon,$$

which is exactly the claimed formula. □

C.2.13 Proof of Proposition B.13

Let $\lambda_1 \geq \cdots \geq \lambda_k > 0$ be eigenvalues of Σ_Y. From $\log\det(\Sigma_Y) \geq k\tau$ we have

$$\prod_{i=1}^{k} \lambda_i \;\geq\; \exp(k\tau).$$

Let $\lambda_{\min} = \lambda_k$. Using $\sum_{i=1}^{k-1} \lambda_i \leq \operatorname{tr}(\Sigma_Y) - \lambda_k \leq m$, the AM–GM inequality yields

$$\prod_{i=1}^{k-1} \lambda_i \;\leq\; \Big(\frac{\sum_{i=1}^{k-1} \lambda_i}{k-1}\Big)^{k-1} \;\leq\; \Big(\frac{m}{k-1}\Big)^{k-1}.$$

Therefore

$$\exp(k\tau) \;\leq\; \prod_{i=1}^{k} \lambda_i = \lambda_k \prod_{i=1}^{k-1} \lambda_i \;\leq\; \lambda_k \Big(\frac{m}{k-1}\Big)^{k-1},$$

so

$$\lambda_{\min}(\Sigma_Y) = \lambda_k \;\geq\; \exp(k\tau)\, \frac{(k-1)^{k-1}}{m^{k-1}}.$$

C.2.14 Proof of Theorem B.14

Let $\lambda_1 \geq \cdots \geq \lambda_k > 0$ be eigenvalues of Σ and $c = \sum_i \lambda_i$. The PR constraint implies

$$\mathrm{PR}(\Sigma) = \frac{(\sum_i \lambda_i)^2}{\sum_i \lambda_i^2} \geq r_0 \quad \Rightarrow \quad \sum_{i=1}^{k} \lambda_i^2 \leq \frac{c^2}{r_0}.$$

Since $\lambda_{\max}^2 = \lambda_1^2 \leq \sum_i \lambda_i^2$, we get

$$\lambda_{\max}(\Sigma) \leq \frac{c}{\sqrt{r_0}}.$$

Next, the log-det constraint implies

$$\prod_{i=1}^{k} \lambda_i = \det(\Sigma) \geq \exp(k\tau).$$

Using $\lambda_i \leq \lambda_{\max}$ for $i = 1, \ldots, k-1$ gives

$$\exp(k\tau) \leq \lambda_{\min}\, \lambda_{\max}^{k-1}, \quad \text{so} \quad \lambda_{\min} \geq \frac{\exp(k\tau)}{\lambda_{\max}^{k-1}} \geq \exp(k\tau) \left(\frac{\sqrt{r_0}}{c}\right)^{k-1}.$$

C.2.15 Proof of Proposition B.16

By Theorem B.4, $\Phi_\phi^{(K)}$ is a C^1 diffeomorphism with $|\det(D\Phi_\phi^{(K)}(s))| = 1$ for all s. For any measurable $\mathcal{U}$ with finite volume, the change-of-variables theorem gives

$$\mathrm{Vol}\big(\Phi_\phi^{(K)}(\mathcal{U})\big) = \int_{\mathcal{U}} \Big|\det(D\Phi_\phi^{(K)}(s))\Big|\, ds = \int_{\mathcal{U}} 1\, ds = \mathrm{Vol}(\mathcal{U}).$$

Hence $\mathrm{Vol}(\mathcal{U}) > 0$ implies $\mathrm{Vol}(\Phi_\phi^{(K)}(\mathcal{U})) > 0$. Therefore the predictor cannot map a positive-volume set to a zero-volume set; any collapse to a lower-volume region must already be present in the encoder outputs.

D Oracle coupling and structural characterization

This appendix proves the additional known-geometry results in Section 3.6. Throughout, $H \in \mathbb{R}^{d\times d}$ is symmetric positive definite and $\alpha = c/d > 0$. The continuous characterization concerns linear one-parameter flows. The exact discrete Gaussian statements concern the quadratic potential $\frac{1}{2} q^\top H q$, rather than an arbitrary learned nonlinear potential.

D.1 Proof of Theorem 3.10

Existence of a constant generator. A continuous one-parameter group of linear maps is automatically differentiable in finite dimensions. Here is a direct argument. For sufficiently small $a > 0$, the matrix

$$B_a := \int_0^a F_s \, ds$$

is invertible, since $a^{-1}B_a \to I$ as $a \downarrow 0$. The group property gives

$$F_t B_a = \int_t^{t+a} F_u \, du.$$

The right-hand side is continuously differentiable in t, with derivative $F_{t+a} - F_t = F_t(F_a - I)$. Therefore

$$F_t' = F_t L, \qquad L := (F_a - I)B_a^{-1},$$

and $F_0 = I$ implies $F_t = e^{tL}$. In particular the resulting linear flow exists for every real t.

The oracle constraints determine the generator. Write

$$L = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \qquad \Gamma_H = \alpha \begin{pmatrix} H^{-1} & 0 \\ 0 & I \end{pmatrix}, \qquad \mathsf{R} = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix}.$$

Differentiating $\mathsf{R}F_t\mathsf{R} = F_{-t}$ at $t = 0$ yields

$$\mathsf{R}L\mathsf{R} = -L.$$

Its diagonal blocks give $A = D = 0$. The kinematic identity $\dot{Q} = P$, required at every state, then fixes $B = I$. Preservation of the centered Gaussian law is equivalent to

$$F_t \Gamma_H F_t^\top = \Gamma_H.$$

Differentiating this identity at zero gives

$$0 = L\Gamma_H + \Gamma_H L^\top = \alpha \begin{pmatrix} 0 & I + H^{-1}C^\top \\ CH^{-1} + I & 0 \end{pmatrix}.$$

Since H is invertible, $C = -H$. This proves uniqueness of

$$L_H = \begin{pmatrix} 0 & I \\ -H & 0 \end{pmatrix}.$$

Conversely, this generator satisfies

$$L_H\Gamma_H + \Gamma_H L_H^\top = 0, \qquad \mathsf{R}L_H\mathsf{R} = -L_H.$$

The first identity implies $\frac{d}{dt}(e^{tL_H}\Gamma_H e^{tL_H^\top}) = 0$, so the Gaussian law is stationary. The second implies $\mathsf{R}e^{tL_H}\mathsf{R} = e^{-tL_H}$, proving reversibility. Its upper block is precisely $\dot{Q} = P$. Finally,

$$L_H = J \begin{pmatrix} H & 0 \\ 0 & I \end{pmatrix}, \qquad \mathcal{H}_H(q,p) = \tfrac{1}{2} q^\top H q + \tfrac{1}{2}\|p\|_2^2.$$

Hence the characterized flow is Hamiltonian and symplectic, although symplecticity was not imposed among the oracle constraints. □

Scope of the characterization. The lifted Gaussian law, its independent fixed-unit momentum, and the momentum-reversal involution are explicit assumptions. They are stronger than equality of the two observable marginals or exchangeability of an observed view pair. Theorem 3.10 does not assert that these observable properties force a unique latent construction. It also does not assert uniqueness among nonlinear vector fields.

The role of reversal can be seen directly within the linear class. If stationarity and $\dot{Q} = P$ are retained but reversal is omitted, the allowed generators are

$$L = \begin{pmatrix} 0 & I \\ -H & D \end{pmatrix}, \qquad D + D^\top = 0.$$

Indeed, the same stationarity calculation fixes the lower-left block to $-H$ and only requires the remaining block to be skew-symmetric. For $d \geq 2$, nonzero such blocks exist. If instead reversal is retained but the fixed-unit kinematic condition is omitted, every generator

$$L_B = \begin{pmatrix} 0 & B \\ -B^\top H & 0 \end{pmatrix}, \qquad B \in \mathbb{R}^{d\times d},$$

preserves the same oracle law and is reversible. Thus the uniqueness claim depends on the declared class and normalization.

D.2 Proof of Corollary 3.11

Set

$$C_t := \cos(tH^{1/2}), \qquad S_t := \sin(tH^{1/2}),$$

using the spectral functional calculus for the symmetric positive definite matrix H. Solving $\ddot{Q} = -HQ$ with initial velocity P_0 gives

$$Q_t = C_t Q_0 + H^{-1/2} S_t P_0, \qquad P_t = -H^{1/2} S_t Q_0 + C_t P_0.$$

Under the oracle law, $Q_0 \sim \mathcal{N}(0, \alpha H^{-1})$ and $P_0 \sim \mathcal{N}(0, \alpha I)$ are independent. Conditioning on $Q_0 = q$ therefore gives

$$Q_t \mid Q_0 = q \sim \mathcal{N}\Big(C_t q,\ \alpha H^{-1/2} S_t^2 H^{-1/2}\Big).$$

This includes degenerate Gaussian conditionals: in an eigenmode where $\sin(t\sqrt{\lambda}) = 0$, the conditional variance in that mode vanishes. At other times and modes the auxiliary momentum produces a nontrivial observable conditional variance.

Write $\Sigma = \alpha H^{-1}$. All the displayed functions of H commute. Using $C_t^2 + S_t^2 = I$ shows

$$\mathrm{Cov}(Q_t) = C_t \Sigma C_t + \alpha H^{-1/2} S_t^2 H^{-1/2} = \Sigma,$$

and independence of Q_0, P_0 gives

$$\mathrm{Cov}(Q_t, Q_0) = C_t \Sigma.$$

Consequently the joint centered Gaussian law of (Q_0, Q_t) has covariance

$$\begin{pmatrix} \Sigma & \Sigma C_t \\ C_t \Sigma & \Sigma \end{pmatrix}.$$

Because $C_t \Sigma = \Sigma C_t$ is symmetric, this covariance is invariant under exchanging the two blocks. The joint law is therefore exchangeable. Finally,

$$M(t) = \mathrm{Cov}(Q_t, Q_0)\Sigma^{-1} = C_t, \qquad M''(0) = -H,$$

which proves the curvature identity. □

What the curvature identity identifies. Knowing the normalized coupling $M(t)$ on a neighborhood of zero determines H through $-M''(0)$. A single normalized coupling matrix $M(t_0)$, by itself, generally does not: in one dimension, ω^2 and $(\omega + 2\pi n/t_0)^2$ give the same cosine for any positive integer n. This observation concerns coupling at one lag, not the full oracle marginal, from which $H = \alpha\Sigma^{-1}$ is already determined when α is known. Neither identity proves that the practical training objective identifies the unknown downstream task geometry.

D.3 Proof of Proposition 3.12

The finite-step map and its invariant law. Let

$$A_h := I - \tfrac{h^2}{2} H, \qquad R_h := I - \tfrac{h^2}{4} H.$$

The stability condition $h^2 \lambda_{\max}(H) < 4$ makes R_h positive definite. For the quadratic potential, a kick–drift–kick leapfrog step has matrix

$$\begin{aligned} F_h &= \begin{pmatrix} I & 0 \\ -\frac{h}{2} H & I \end{pmatrix} \begin{pmatrix} I & hI \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ -\frac{h}{2} H & I \end{pmatrix} \\ &= \begin{pmatrix} A_h & hI \\ -hHR_h & A_h \end{pmatrix}. \end{aligned}$$

The factors A_h, R_h, H commute and satisfy

$$A_h^2 + h^2 H R_h = I.$$

For $\Gamma_{H,h} = \alpha \operatorname{diag}(H^{-1}, R_h)$, direct multiplication now gives

$$F_h \Gamma_{H,h} F_h^\top = \alpha \begin{pmatrix} H^{-1}(A_h^2 + h^2 H R_h) & 0 \\ 0 & R_h(A_h^2 + h^2 H R_h) \end{pmatrix} = \Gamma_{H,h}.$$

Thus every iterate preserves the centered Gaussian law. Its density is the quadratic shadow Gibbs law

$$\pi_{H,h}(q,p) \propto \exp\left[-\frac{1}{2\alpha} \left(q^\top H q + p^\top R_h^{-1} p \right) \right]. \tag{45}$$

Both quadratic forms are positive definite, so this density is normalizable. The associated quadratic energy is exactly conserved by F_h. No assertion that this particular invariant energy generates F_h as its exact time-h flow is needed.

Chebyshev form of the coupling. Let $\mathcal{A}_h = \operatorname{diag}(A_h, A_h)$. Another direct multiplication gives

$$F_h^2 - 2\mathcal{A}_h F_h + I_{2d} = 0.$$

Moreover F_h commutes with $\mathcal{A}_h$. If B_K denotes the upper-left block of F_h^K, then

$$B_0 = I, \qquad B_1 = A_h, \qquad B_{K+1} = 2A_h B_K - B_{K-1}.$$

This is the defining recurrence of the first-kind Chebyshev polynomials, so

$$B_K = T_K(A_h).$$

The upper-right block is similarly $hU_{K-1}(A_h)$ for $K \geq 1$, where U_j is the second-kind Chebyshev polynomial. In particular,

$$Q_K = T_K(A_h) Q_0 + h U_{K-1}(A_h) P_0.$$

Under $\pi_{H,h}$, the two initial coordinates are independent and $\operatorname{Cov}(Q_0) = \alpha H^{-1}$. Therefore

$$\operatorname{Cov}(Q_K, Q_0) = T_K(A_h)\, \alpha H^{-1},$$

which proves (25).

The invariant marginal also gives the exact projected conditional law

$$Q_K \mid Q_0 = q \sim \mathcal{N}\big(T_K(A_h) q,\, \alpha H^{-1}\big(I - T_K(A_h)^2\big)\big). \tag{46}$$

One can obtain the covariance either by Gaussian conditioning or from

$$1 - T_K(x)^2 = (1 - x^2) U_{K-1}(x)^2$$

and $I - A_h^2 = h^2 H R_h$. Since the spectrum of A_h lies in $(-1, 1)$, the covariance is positive semidefinite. As in the continuous proof, its stationary Gaussian pair is exchangeable because $T_K(A_h)$ is symmetric and commutes with H^{-1}.

The finite-step correction is essential. The preserved law $\pi_{H,h}$ retains the original oracle content marginal but changes the momentum covariance from αI to αR_h. If instead one initializes with the original full oracle covariance $\Gamma_H = \alpha \operatorname{diag}(H^{-1}, I)$, then already after one step,

$$\operatorname{Cov}(Q_1) = \alpha A_h H^{-1} A_h + \alpha h^2 I = \alpha H^{-1} + \frac{\alpha h^4}{4} H.$$

For $h > 0$, this differs from αH^{-1}. Thus exact preservation of the original full oracle lift is not a property of finite-step leapfrog. An alternative invariant covariance is $\alpha \operatorname{diag}((H R_h)^{-1}, I)$; that choice preserves the momentum normalization while modifying the content covariance. The proposition uses $\Gamma_{H,h}$ so that the Section 3 content marginal stays explicit.

Continuous limit and structured locality. For an eigenvalue $\lambda > 0$ of H, write

$$\theta_h(\lambda) := \arccos(1 - h^2\lambda/2).$$

Then $\theta_h(\lambda)/h \to \sqrt{\lambda}$. Hence, if $h \to 0$ and $Kh \to t$,

$$T_K(1 - h^2\lambda/2) = \cos(K\theta_h(\lambda)) \longrightarrow \cos(t\sqrt{\lambda}).$$

The finite-dimensional spectral decomposition gives

$$T_K(A_h) \longrightarrow \cos(tH^{1/2}), \qquad R_h \longrightarrow I,$$

and the conditional law in (46) converges to the continuous oracle conditional.

If a graph records the nonzero off-diagonal entries of H, then $(H^j)_{uv} = 0$ whenever the graph distance between u and v exceeds j. This follows by expanding the matrix product: a nonzero summand requires a walk of at most j off-diagonal edges. Since $T_K(I - h^2H/2)$ is a polynomial in H of degree at most K, its (u, v) entry vanishes when that distance exceeds K. This locality statement concerns a sparse quadratic operator H; it does not assert that the learned nonlinear potential in the experiments has this sparsity. □

Reported numerical oracle checks. Numerical checks reported during review covered 18 structured SPD cases, including non-degenerate observable conditionals, relative H-recovery error below 8.1×10^{-8}, and second-order shadow convergence. The shadow precision used was $H_h = H(I - h^2H/4)$, corresponding to the alternative invariant covariance above. These known-geometry consistency checks do not estimate the downstream operator or the price of isotropy for image representations.

D.4 Oracle compatibility and downstream performance

Proposition D.1 (Oracle defect of a rotated force)**.** *For $U \in \mathbb{R}^{d\times d}$, consider*

$$\dot{Q} = P, \qquad \dot{P} = -UHQ, \qquad L_U = \begin{pmatrix} 0 & I \\ -UH & 0 \end{pmatrix}.$$

The resulting continuous linear flow preserves $\mathcal{N}(0, \Gamma_H)$ if and only if $U = I$. It is nevertheless volume-preserving and reversible under R for every U.

Proof. The stationarity defect is

$$L_U\Gamma_H + \Gamma_H L_U^\top = \alpha \begin{pmatrix} 0 & I - U^\top \\ I - U & 0 \end{pmatrix}.$$

If the flow preserves the oracle covariance, differentiation at zero forces this matrix to vanish, hence $U = I$. The converse is Theorem 3.10. For every U, $\operatorname{tr}(L_U) = 0$, so $\det(e^{tL_U}) = e^{t\operatorname{tr}(L_U)} = 1$. The off-diagonal block form also gives $\mathsf{R}L_U\mathsf{R} = -L_U$, and hence reversibility. □

Proposition D.2 (Equal task performance despite oracle incompatibility)**.** *Let $d = 3$, $H = I_3$, and*

$$U = \operatorname{diag}(1, U_2),$$

where U_2 is a two-dimensional rotation with angle $\vartheta \in (0, \pi)$. Initialize both the Hamiltonian and rotated-force systems from the same oracle law and define

$$Y = \mathbf{1}[Q_{0,1} > 0].$$

For every fixed t, the two systems have the same Bayes-optimal classification error for predicting Y from Q_t, although the rotated-force flow does not preserve the full oracle law.

Proof. In both systems the first coordinate obeys

$$\ddot{Q}_1 = -Q_1, \qquad Q_{t,1} = (\cos t)Q_{0,1} + (\sin t)P_{0,1}.$$

Its joint law with Y is therefore identical. The remaining coordinates evolve within the separate $(2, 3)$ block. Their initial coordinates are independent of $(Q_{0,1}, P_{0,1})$, and this independence is preserved by the deterministic blockwise evolution. Consequently,

$$\mathbb{P}(Y = 1 \mid Q_t) = \mathbb{P}(Y = 1 \mid Q_{t,1})$$

in each system, and the conditional law on the right is the same in both. The Bayes classification error is thus the same. Since $U \neq I$, Proposition D.1 proves failure of full oracle stationarity. □

This example separates two assertions. The oracle theorem characterizes which linear coupling satisfies all its assumptions. It does not imply that violating an assumption must worsen every downstream task. The example is a population statement about a specified observable task, not a claim that the training dynamics or finite-sample scores of the two learned models must coincide. Accordingly, the matched controls restrict empirical causal attribution without contradicting the conditional oracle characterization.

D.5 Canonical form under cotangent coordinate changes

Proposition D.3 (Invariant constant alternating form)**.** *Let $\Omega \in \mathbb{R}^{2d\times 2d}$ be skew-symmetric. Suppose that for every $B \in GL(d)$,*

$$S_B^\top \Omega S_B = \Omega, \qquad S_B = \begin{pmatrix} B & 0 \\ 0 & B^{-\top} \end{pmatrix}.$$

Then $\Omega = \beta J$ for some $\beta \in \mathbb{R}$. The form is nondegenerate if and only if $\beta \neq 0$.

Proof. Write

$$\Omega = \begin{pmatrix} E & G \\ -G^\top & F \end{pmatrix}, \qquad E^\top = -E, \quad F^\top = -F.$$

Taking $B = rI$ with $r > 0$ gives

$$S_B^\top \Omega S_B = \begin{pmatrix} r^2 E & G \\ -G^\top & r^{-2} F \end{pmatrix}.$$

Invariance for, say, $r = 2$ forces $E = F = 0$. The upper-right block of the general invariance identity then requires

$$B^\top G B^{-\top} = G,$$

or $B^\top G = G B^\top$, for every invertible B. Commutation with all invertible diagonal matrices forces G to be diagonal. Commutation with all coordinate permutation matrices then forces its diagonal entries to agree. Thus $G = \beta I$ and $\Omega = \beta J$. Conversely $S_B^\top J S_B = J$ by direct multiplication. Finally, J is invertible, so βJ is nondegenerate exactly when $\beta \neq 0$. □

The proposition concerns constant alternating forms and the declared cotangent action $(q, p) \mapsto (Bq, B^{-\top}p)$. It does not characterize arbitrary nonlinear predictors. Nor does it make the fixed-unit kinetic normalization invariant under all such coordinate changes. Indeed, writing $q' = Bq$, $p' = B^{-\top}p$ transforms $\frac{1}{2}\|p\|_2^2$ into $\frac{1}{2}p'^\top BB^\top p'$, and $\dot{q} = p$ into $\dot{q}' = BB^\top p'$. The canonical two-form is preserved, while the kinetic metric changes. The unit kinetic metric used in the oracle theorem and implementation is therefore a condition in the chosen coordinate chart.

E Supplementary theory: maximum entropy on phase space and symplectic factorization

Theorem E.1 (Maximum entropy on phase space under an expected energy constraint)**.** *Let $\mathcal{H} : \mathbb{R}^m \to \mathbb{R}$ be measurable and assume there exists $\beta > 0$ such that the partition function*

$$Z(\beta) := \int_{\mathbb{R}^m} e^{-\beta \mathcal{H}(s)}\, ds$$

is finite. Fix $c \in \mathbb{R}$ and consider the set of densities

$$\mathcal{P}_c := \left\{ p \text{ density on } \mathbb{R}^m : \int p(s)\, ds = 1,\ \int \mathcal{H}(s) p(s)\, ds = c,\ h(p) > -\infty \right\},$$

where $h(p) := -\int p \log p$ is differential entropy. If there exists $\beta^\star > 0$ such that $\mathbb{E}_{p_{\beta^\star}}[\mathcal{H}(S)] = c$ for

$$p_\beta(s) = \frac{1}{Z(\beta)} e^{-\beta \mathcal{H}(s)},$$

then the unique maximizer of $h(p)$ over $\mathcal{P}_c$ is $p^\star = p_{\beta^\star}$.

Proof. Fix $\beta > 0$ with $Z(\beta) < \infty$ and define $p_\beta(s) = Z(\beta)^{-1} e^{-\beta \mathcal{H}(s)}$. For any $p \in \mathcal{P}_c$,

$$\mathrm{KL}(p \| p_\beta) = \int p(s) \log \frac{p(s)}{p_\beta(s)} \, ds \geq 0,$$

with equality iff $p = p_\beta$ a.e. Expanding $\log p_\beta(s) = -\beta \mathcal{H}(s) - \log Z(\beta)$ gives

$$\begin{aligned}
\mathrm{KL}(p \| p_\beta) &= \int p \log p \, ds - \int p \log p_\beta \, ds \\
&= -h(p) - \int p(s) \big(-\beta \mathcal{H}(s) - \log Z(\beta) \big) \, ds \\
&= -h(p) + \beta \underbrace{\int \mathcal{H}(s) p(s) \, ds}_{=c} + \log Z(\beta) \underbrace{\int p(s) \, ds}_{=1} \\
&= -h(p) + \beta c + \log Z(\beta).
\end{aligned}$$

Thus $h(p) \leq \beta c + \log Z(\beta)$ for all $p \in \mathcal{P}_c$, with equality iff $p = p_\beta$.

If there exists $\beta^\star$ such that $p_{\beta^\star} \in \mathcal{P}_c$ (i.e., it matches the energy constraint), then $p_{\beta^\star}$ attains the upper bound and is therefore a maximizer. Uniqueness follows from the equality condition for KL divergence. □

Corollary E.2 (Energy-based densities are invariant under Hamiltonian flow)**.** *Let $\mathcal{H} \in C^2(\mathbb{R}^{2d_0})$ and let Φ_t be the Hamiltonian flow of $\dot{s} = J \nabla \mathcal{H}(s)$, assumed to be a C^1 diffeomorphism for the times of interest. If S has density $p(s) = g(\mathcal{H}(s))$ for some measurable $g : \mathbb{R} \to \mathbb{R}_+$, then for all t, $\Phi_t(S) \overset{d}{=} S$. In particular, the Gibbs density $p(s) \propto e^{-\beta \mathcal{H}(s)}$ is invariant.*

Proof. Energy is conserved along Hamiltonian trajectories: if $s(t) = \Phi_t(s_0)$, then

$$\frac{d}{dt} \mathcal{H}(s(t)) = \nabla \mathcal{H}(s(t))^\top \dot{s}(t) = \nabla \mathcal{H}(s(t))^\top J \nabla \mathcal{H}(s(t)) = 0,$$

since $J^\top = -J$ implies $u^\top J u = 0$ for all u. Thus $\mathcal{H}(\Phi_t(s)) = \mathcal{H}(s)$.

Also Hamiltonian flows are volume-preserving (Liouville): $|\det(D\Phi_t(s))| = 1$ for all s, t. Let $Y = \Phi_t(S)$. By change of variables,

$$p_Y(y) = p(\Phi_{-t}(y)) \cdot \left| \det \big(D\Phi_{-t}(y) \big) \right| = g(\mathcal{H}(\Phi_{-t}(y))) \cdot 1 = g(\mathcal{H}(y)) = p(y).$$

Hence Y and S have the same density. □

Proposition E.3 (Generic kick–drift–scaling factorization of linear symplectic maps)**.** *Let $A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{Sp}(2d_0)$ and assume the block $d \in \mathbb{R}^{d_0 \times d_0}$ is invertible. Define*

$$B := b d^{-1}, \qquad C := d^{-1} c.$$

Then B and C are symmetric and

$$A = \underbrace{\begin{pmatrix} I & B \\ 0 & I \end{pmatrix}}_{\text{upper shear (drift)}} \underbrace{\begin{pmatrix} d^{-\top} & 0 \\ 0 & d \end{pmatrix}}_{\text{symplectic scaling}} \underbrace{\begin{pmatrix} I & 0 \\ C & I \end{pmatrix}}_{\text{lower shear (kick)}}.$$

Proof. Since $A \in \mathrm{Sp}(2d_0)$, it satisfies $A^\top J A = J$, which implies the standard block identities $b^\top d = d^\top b$ and $a^\top d - c^\top b = I$. From $b^\top d = d^\top b$ and invertibility of d,

$$B^\top = (b d^{-1})^\top = d^{-\top} b^\top = b d^{-1} = B,$$

so B is symmetric.

Similarly, $A J A^\top = J$ also holds for symplectic A, and its block identities include $c d^\top = d c^\top$. Then

$$C^\top = (d^{-1} c)^\top = c^\top d^{-\top} = d^{-1} c = C,$$

so C is symmetric.

Now multiply the three proposed factors:

$$\begin{pmatrix} I & B \\ 0 & I \end{pmatrix} \begin{pmatrix} d^{-\top} & 0 \\ 0 & d \end{pmatrix} \begin{pmatrix} I & 0 \\ C & I \end{pmatrix} = \begin{pmatrix} d^{-\top} + BdC & Bd \\ dC & d \end{pmatrix}.$$

The bottom-right block matches d. The top-right block matches b because $Bd = (bd^{-1})d = b$. The bottom-left block matches c because $dC = d(d^{-1}c) = c$.

For the top-left block, use the symplectic identity $ad^\top - bc^\top = I$ (from $AJA^\top = J$) and right-multiply by $d^{-\top}$:

$$a - bc^\top d^{-\top} = d^{-\top}.$$

Using $c^\top d^{-\top} = (d^{-1}c)^\top = C^\top = C$, this becomes

$$a = d^{-\top} + bC = d^{-\top} + (Bd)C = d^{-\top} + BdC,$$

which matches the computed top-left block. Hence the factorization holds. □

F Further results and figures

We refer to the method uniformly as *HamJEPA*; in the code/config files, the suffix `hjepa_mv` simply denotes the minimal multi-view implementation used in our experiments.

F.1 Frozen-feature diagnostic summaries

Figures 1 (CIFAR-100) and 3 (ImageNet-100) summarize the geometry and non-parametric usefulness of *frozen* encoder representations for the SIGReg baseline and HamJEPA latent states. Each method panel contains four diagnostics computed on *raw* encoder features (i.e., no whitening or other post-processing):

1. k**NN sweep:** top-1 accuracy as a function of neighborhood size k.
2. **Covariance eigenspectrum:** top-256 eigenvalues (log-scale) of the *mean-centered* empirical covariance.
3. **Random-pair cosine similarities:** histogram of cosine similarities after per-vector ℓ_2 normalization *without* mean subtraction.
4. **Feature norms:** histogram of ℓ_2 feature norms.

A key subtlety is that the covariance spectrum is computed after *mean-centering*, whereas the cosine histogram is computed on normalized features *without* centering. Consequently, strongly positive random-pair cosines can arise from a dominant *mean direction* even when the mean-centered covariance remains high-rank. We exploit this complementarity in the discussions below.

F.2 Discussion CIFAR-100

F.2.1 30 epoch results

These results are for the 30 epoch pretraining.

(1) Neighborhood quality: k**NN sweeps.** In Fig. 1a, the SIGReg baseline peaks at small k and degrades as k grows, consistent with semantics being locally present but becoming less class-pure as neighborhoods expand. Quantitatively, SIGReg peaks at $27.46 \pm 0.31\%$ (at k=1) and falls to $23.10 \pm 0.32\%$ (at k=200).

HamJEPA improves the entire sweep for all latent variants in Fig. 1. Notably, the *single* q state (1024-D) outperforms the SIGReg baseline (2048-D) across the plotted k values: $31.45 \pm 0.27\%$ at k=20 for q, versus $26.56 \pm 0.18\%$ for SIGReg at k=20 (a $+4.88 \pm 0.11$ point gain). The p and (q, p) variants achieve $29.71 \pm 0.25\%$ and $30.88 \pm 0.39\%$ at k=20, respectively. That (q, p) does not exceed q suggests that, under the default weighting, p contributes less additional *metric-consistent* neighborhood structure than q, even though it remains informative for linear decoding (see below).

(2) Linear probe vs. kNN: separability without neighborhood purity. While kNN distinguishes the latent variants, the linear probe largely does not: SIGReg reaches $30.43 \pm 0.30\%$ top-1, whereas HamJEPA reaches $33.95 \pm 0.24\%$, $33.07 \pm 0.36\%$ and $34.18 \pm 0.17\%$ top-1 for q, p, and (q, p) respectively. A useful interpretation is that p is *linearly* discriminative but induces less clean neighborhoods under the raw metric, so its gains show up more clearly in the probe than in kNN.

(3) Covariance eigenspectrum: rank proxies and collapse checks. The top-right panels show mean-centered covariance spectra (top 256 eigenvalues, log-scale). A convenient summary is the effective-rank proxy computed from these eigenvalues: SIGReg has effective rank $\approx 109.73 \pm 0.61$ (top-256), whereas HamJEPA-q increases to 125.58 ± 3.33. In contrast, HamJEPA-p is more concentrated (effective rank 79.46 ± 3.99), and (q, p) lies in between (93.10 ± 4.00). Two points follow: (i) HamJEPA-q does *not* achieve its downstream gain via low-rank collapse (the centered covariance is high-rank), and (ii) the q/p split is geometrically meaningful: q and p allocate variance differently even when their linear probe scores are similar.

(4) Cosine similarities: diagnosing mean-dominance. The bottom-left panels reveal the largest qualitative difference. SIGReg's random-pair cosines are centered near zero (mean 0.01699 ± 0.00089) with substantial negative mass, which is consistent with its isotropy-encouraging design.

HamJEPA's cosines shift sharply positive: mean cosine is 0.71561 ± 0.00129 for q, 0.61687 ± 0.00409 for p, and 0.66613 ± 0.00270 for (q, p). Crucially, this strong alignment coexists with a *high-rank mean-centered covariance* (especially for q). This combination is most naturally explained by a large *non-zero mean feature direction* that dominates dot products when features are not explicitly centered: the covariance spectrum removes that mean direction by construction, while the cosine histogram does not. In other words, the CIFAR-100 HamJEPA embeddings look "cone-like" in raw normalized space primarily due to mean-dominance rather than variance collapse.

(5) Feature norms: scale control and the (q, p) energy split. SIGReg feature norms are concentrated around 44.891 ± 0.090, whereas HamJEPA's q and p have smaller typical norms around 31.358 ± 0.017 and 31.281 ± 0.029. The concatenated (q, p) norm shifts back to 44.303 ± 0.027, closely matching the Pythagorean combination $\|[q; p]\|_2 \approx \sqrt{\|q\|_2^2 + \|p\|_2^2}$, which is what one expects if q and p maintain comparable energy rather than one collapsing to near-zero. This provides a simple consistency check that the phase-space parameterization retains a balanced contribution from both coordinates.

F.2.2 Original 80-epoch single-run results

The following figures, table, and diagnostics retain the original single-run comparison. The replicated controls are reported separately in Appendix F.4.

MLP-HJEPA keeps the phase-space encoder, channel-wise q/p split, identity projector, prediction loss, anti-collapse regularizers, optimizer, batch size, and training schedule fixed, and replaces only the symplectic leapfrog map with a non-symplectic residual MLP predictor. This ablation is intentionally strong: MLP-HJEPA reaches high linear accuracy (42.26–42.77), showing that the phase-space split and regularizers can produce linearly useful features. However, its kNN@20 accuracy remains essentially at the SIGReg level (28.10–28.30 vs. 27.98). By contrast, HamJEPA with the symplectic predictor reaches 34.43 kNN@20 and 44.59 linear accuracy using the q readout. In this single-run comparison, replacing the MLP by the Hamiltonian predictor yields a $+6.33$ point kNN@20 gain over MLP-HJEPA-q and a $+6.45$ point gain over full-dimensional SIGReg at the same 80-epoch budget. The matched controls below separate this predictor-family comparison from claims about individual structural properties.

By design, the CIFAR-100 pretraining curves tend to plateau because the experiment is set up as a minimal, controlled training regime, whose goal is to isolate representational geometry effects rather than to chase absolute accuracy with an aggressively optimized recipe. CIFAR-100 is small and low-resolution, so the encoder typically learns the dominant semantic invariances early; after that point, additional epochs mainly yield diminishing returns (small refinements in neighborhood structure or feature scaling) that often do not translate into large gains in linear-probe/(k)NN top-1 under a fixed evaluation protocol. In other words, once both methods reach a "good-enough" feature space, the remaining headroom is mostly governed by recipe choices we intentionally avoid here

(stronger augmentation, longer schedules, larger models, distillation/EMA, more hyperparameter search), and their absence limits the external validity of these results.

Method	Variant	Linear probe (%)	kNN@20 (%)	kNN best (%)
SIGReg	single	33.95	27.98	28.71 @ k=1
MLP-HJEPA (non-symplectic)	q	42.26	28.10	28.10 @ k=20
MLP-HJEPA (non-symplectic)	p	41.90	28.30	28.30 @ k=20
MLP-HJEPA (non-symplectic)	(q, p)	42.77	28.15	28.15 @ k=20
HamJEPA (symplectic)	q	44.59	34.43	34.43 @ k=20
HamJEPA (symplectic)	p	44.44	32.96	32.96 @ k=20
HamJEPA (symplectic)	(q, p)	44.52	33.66	33.66 @ k=20

Table 4: **CIFAR-100, original single-run 80-epoch predictor-family ablation.** MLP-HJEPA keeps the phase-space encoder, channel-wise q/p split, identity projector, prediction loss, anti-collapse regularizers, optimizer, batch size, and 80-epoch schedule fixed, and replaces only the symplectic leapfrog predictor by a non-symplectic residual MLP predictor.

Predictor-family ablation: what changes and what stays fixed. In addition to SIGReg and symplectic HamJEPA, we evaluate a non-symplectic predictor ablation, *MLP-HJEPA*, that keeps the phase-space encoder, channel-wise q/p split, identity projector, prediction loss, anti-collapse regularizers, optimizer, batch size, and training schedule fixed, and replaces only the leapfrog Hamiltonian rollout by a generic residual MLP rollout predictor. Because Fig. 2 visualizes only SIGReg and symplectic HamJEPA, we summarize the MLP ablation numerically in Table 4. This comparison changes the predictor family and does not separately isolate symplecticity, reversibility, or volume preservation.

(1) Neighborhood quality: kNN sweeps and local semantic structure. At 80 epochs, SIGReg reaches 27.98% at kNN@20 and peaks at 28.71% over the full sweep (at k=1). It peaks at small k and degrades as k grows, consistent with semantics being locally present but becoming less class-pure as neighborhoods expand. The non-symplectic MLP-HJEPA ablation barely changes this picture: its best kNN@20 is only 28.30% (from the p readout), a gain of just $+0.32$ points over SIGReg, and its best-over-k score is in fact slightly *below* SIGReg (28.30% vs. 28.71%). By contrast, symplectic HamJEPA is qualitatively different: the q readout reaches 34.43% at kNN@20 and also attains the best-over-k value of 34.43% (at k=20), namely gains of $+6.45$ points at kNN@20 and $+5.72$ points best-over-k relative to SIGReg. Notably, the *single* q state (1024-D) outperforms the SIGReg baseline (2048-D) across the plotted k values. Best-vs.-best, the symplectic predictor outperforms the MLP predictor by $+6.13$ points at kNN@20. This is evidence of a predictor-family difference in neighborhood quality; the matched controls in Appendix F.4 do not identify symplecticity as its cause.

(2) Linear probe: phase-space structure and predictor-family comparison. The linear-probe comparison is more nuanced. MLP-HJEPA already substantially outperforms SIGReg, reaching 42.77% top-1 in its best variant ((q, p)) versus 33.95% for SIGReg, a gain of $+8.82$ points. Symplectic HamJEPA pushes this further to 44.59% (best q readout), i.e. another $+1.82$ points over MLP-HJEPA and $+10.64$ over SIGReg. This shows that the *phase-space parameterization plus encoder-side anti-collapse bridge* can give linearly useful features with a non-symplectic predictor; it does not isolate the contribution of each component. However, once nonparametric neighborhood quality is examined, the predictor family becomes decisive: the MLP predictor narrows much of the probe gap, but essentially none of the kNN gap.

(3) q/p decomposition: structured phase space versus generic feature blocks. Within MLP-HJEPA, the three readouts are almost indistinguishable: linear probe is 42.26%/41.90%/42.77% for q/p/(q, p), and kNN@20 is 28.10%/28.30%/28.15%. In this MLP comparison, the two blocks have similar downstream scores. This contrasts with symplectic HamJEPA, where q is the cleanest neighborhood readout, p is slightly weaker, and (q, p) remains close but does not dominate. These readout differences are descriptive and do not establish that symplecticity causes functional differentiation.

(4) Mean-centered covariance geometry: high-rank symplectic q versus concentrated MLP states. The scalar covariance diagnostics sharpen this distinction. SIGReg has effective rank ≈ 99.80 and participation ratio ≈ 58.32 on the top-256 eigenspectrum. Symplectic HamJEPA-q is substantially broader still, with effective rank ≈ 128.98 and participation ratio ≈ 70.34, confirming that its downstream gains are *not* coming from collapse onto a narrow subspace. By contrast, MLP-HJEPA is much more spectrally concentrated: its effective rank is only ≈ 42.06 for q, ≈ 43.95 for p, and ≈ 42.85 for (q, p), with participation ratios only ≈ 23.65–24.99. So although the MLP ablation yields good linear-probe accuracy, it does so with a much more concentrated mean-centered covariance geometry than symplectic HamJEPA. Put differently: the MLP predictor can produce a linearly usable representation, but it does not organize a broad, high-rank content geometry in the way the symplectic predictor does.

(5) Cosine similarities and feature norms: the predictor changes geometry, not just scale. The random-pair cosine statistics again place MLP-HJEPA much closer to SIGReg than to symplectic HamJEPA. SIGReg has cosine mean ≈ 0.006 with standard deviation ≈ 0.121, while MLP-HJEPA remains near zero as well (cosine means ≈ 0.042 for q, ≈ 0.053 for p, and ≈ 0.047 for (q, p), all with standard deviation around 0.19). Symplectic HamJEPA, in contrast, exhibits strongly positive cosine means (≈ 0.625 for q, ≈ 0.443 for p, and ≈ 0.534 for (q, p)), reflecting the markedly more aligned raw geometry visible in Fig. 2. Importantly, this difference is *not* explained by trivial scale effects: the feature norms of MLP-HJEPA are nearly the same as those of symplectic HamJEPA (means ≈ 32.21, 31.89, and 45.32 for MLP-HJEPA versus ≈ 31.81, 31.88, and 45.06 for symplectic HamJEPA). Thus the predictor family is primarily changing *how* variance is organized and aligned, not simply how large the features are.

Takeaway (CIFAR-100, 80 epochs). The MLP ablation shows that the phase-space encoder, q/p split, and non-isotropic anti-collapse regularization can produce linearly useful representations with a non-symplectic predictor. In this original comparison, HamJEPA produces much better kNN structure, a broader and less concentrated mean-centered covariance for the content coordinate, and a more distinctive raw embedding geometry. The replicated force-rotation and non-reversible controls in Appendix F.4 retain the gain, so neither symplecticity nor reversibility is identified as individually necessary.

F.3 Discussion ImageNet-100

(1) Frozen-feature quality: kNN and probe. On ImageNet-100, HamJEPA again improves non-parametric and linear evaluation. In Fig. 3, the dimension-matched comparison (2048-D) is SIGReg (q, p) versus HamJEPA (q, p). At the final checkpoint, SIGReg (q, p) reaches a best kNN top-1 of 21.7% (peak near k=50), whereas HamJEPA (q, p) reaches 25.9% (a $+4.2$ point gain). For linear probing, SIGReg (q, p) reaches 25.5% top-1, whereas HamJEPA (q, p) reaches 32.0% (a $+6.5$ point gain).

Figure 5 additionally provides a strict 1024-D comparison on q: at epoch 45, HamJEPA-q improves linear probe by $+7.52$ points and best kNN by $+4.90$ points over SIGReg-q (as annotated).

(2) Covariance rank: HamJEPA spreads variance across many directions. The covariance spectra (top-right) and the effective-rank proxy show a stark contrast: SIGReg-q and SIGReg-p have low effective rank ($\approx$ 38–39 on the top-256 eigenvalues), whereas HamJEPA-q and HamJEPA-p are much higher ($\approx$ 95–98). Thus, on ImageNet-100, HamJEPA's gains are associated with *more* variance being spread across principal directions (higher rank), not less.

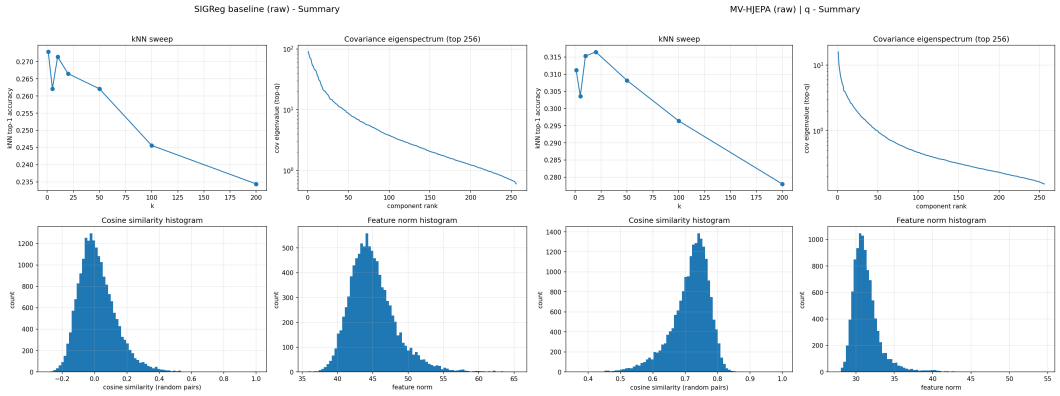
(3) Cosine similarities: mild global alignment without extreme cone structure. Unlike CIFAR-100, ImageNet-100 does *not* show an extreme cone effect for HamJEPA. SIGReg's random-pair cosines remain centered near zero (mean ≈ 0), while HamJEPA shifts to moderately positive cosines (mean ≈ 0.27–0.28 depending on $q/p/(q, p)$). Given the simultaneously high effective rank of the mean-centered covariance, this again suggests a shared direction and/or non-zero mean, but the effect is substantially weaker than on CIFAR-100.

(4) q/p decomposition: complementarity for SIGReg, redundancy for HamJEPA. Figure 4 makes the structural difference explicit. For SIGReg+tokens, (q, p) is consistently best throughout

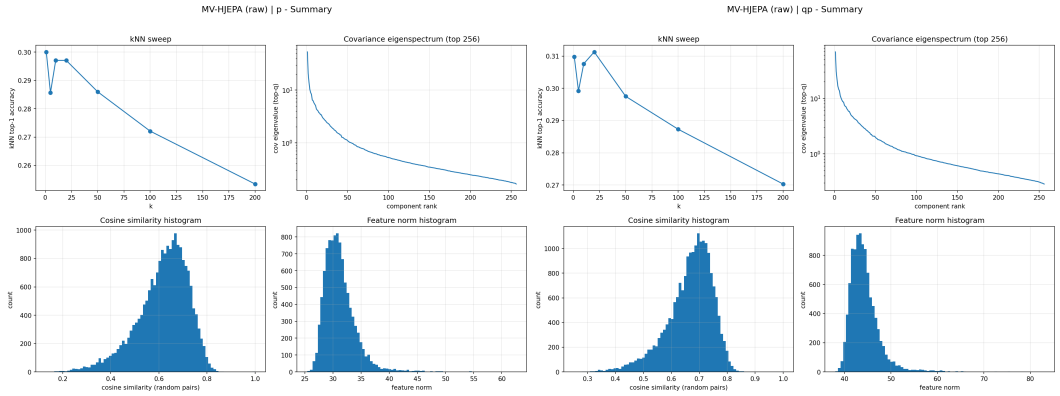
training, indicating that q and p halves are not individually sufficient but become competitive when concatenated. For HamJEPA, q and p are individually as strong as (q, p) (both for probe and kNN), suggesting that both coordinates carry similar semantic content and that concatenation adds little.

(5) Dynamics: accuracy tracks geometry for HamJEPA, not for SIGReg. In Fig. 5, HamJEPA's effective rank increases over training while probe accuracy improves, yielding a strong positive association in the probe-versus-rank plot (annotated ρ). SIGReg shows the opposite: effective rank decreases while accuracy rises slightly, yielding a negative association. This supports the interpretation that HamJEPA's ImageNet gains co-occur with a progressively less concentrated (mean-centered) covariance structure.

Takeaway (ImageNet-100). On ImageNet-100, HamJEPA improves kNN and linear probing while producing substantially higher-rank mean-centered covariance than SIGReg and only moderately more positive cosine alignment. In contrast to SIGReg+tokens, HamJEPA makes both q and p individually strong, with little benefit from concatenation.

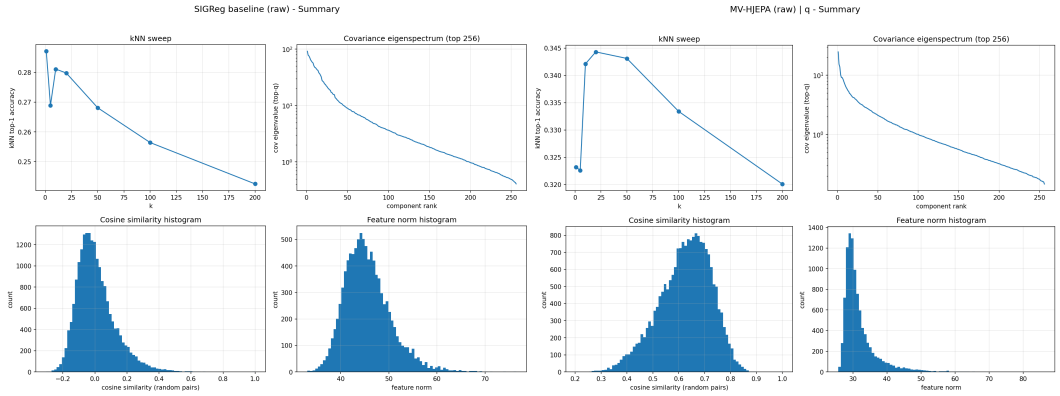


(a) SIGReg (raw). (b) HamJEPA q (raw).



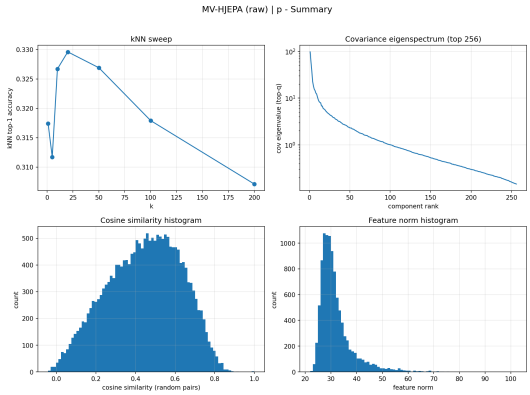
(c) HamJEPA p (raw). (d) HamJEPA (q, p) (raw).

Figure 1: **Frozen-feature diagnostic summaries (raw) 30 epochs.** Each panel contains: (top-left) kNN sweep; (top-right) covariance eigenspectrum (top 256, log scale); (bottom-left) cosine similarity histogram (random pairs); (bottom-right) feature norm histogram.

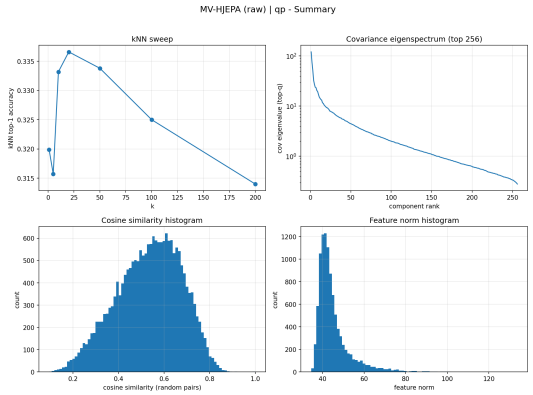


(a) SIGReg (raw).

(b) HamJEPA q (raw).

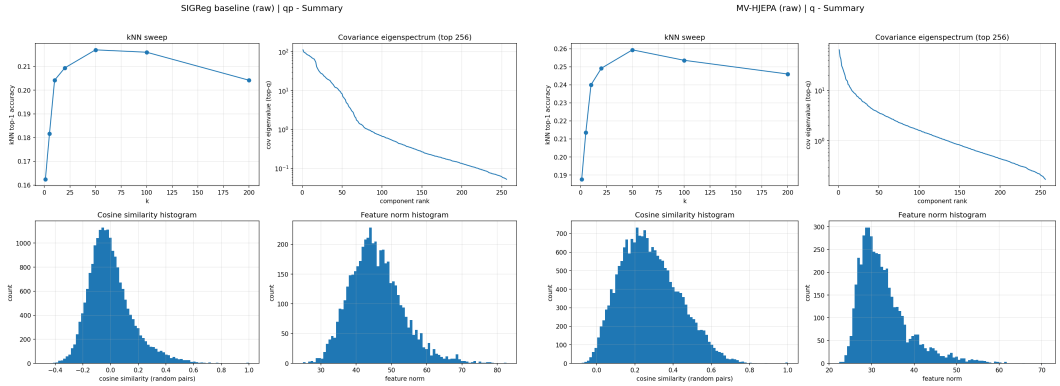


(c) HamJEPA p (raw).



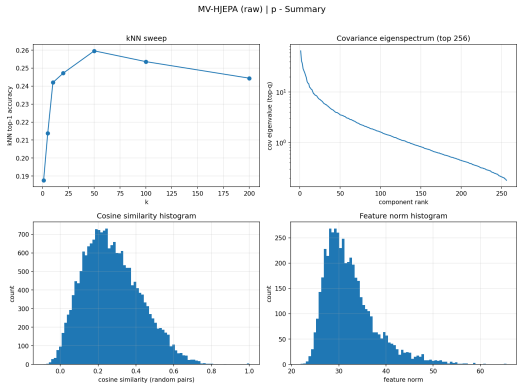
(d) HamJEPA (q, p) (raw).

Figure 2: **CIFAR-100 frozen-feature diagnostic summaries (raw), original single-run 80-epoch results.** Each panel contains: (top-left) kNN sweep; (top-right) mean-centered covariance eigenspectrum (top 256, log scale); (bottom-left) random-pair cosine similarity histogram; (bottom-right) feature norm histogram.

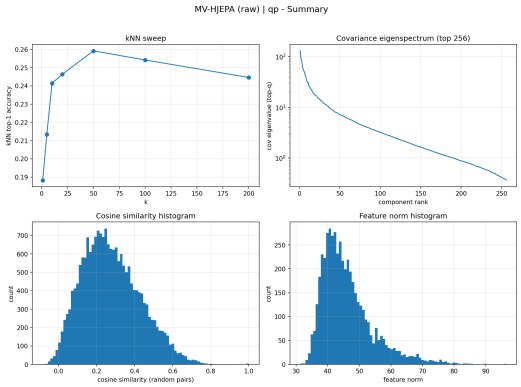


(a) SIGReg (raw).

(b) HamJEPA q (raw).

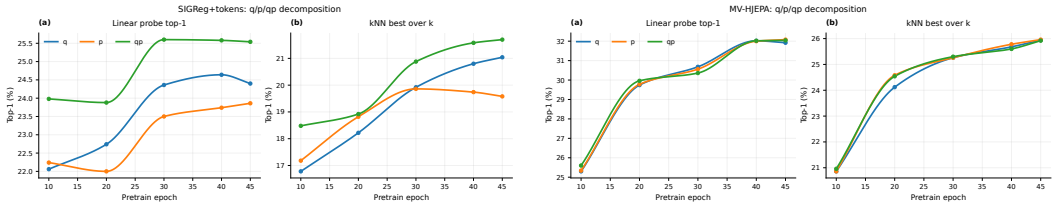


(c) HamJEPA p (raw).



(d) HamJEPA (q, p) (raw).

Figure 3: **ImageNet-100 frozen-feature diagnostic summaries (raw).** Each panel contains: (top-left) kNN sweep; (top-right) covariance eigenspectrum (top 256, log scale); (bottom-left) cosine similarity histogram (random pairs); (bottom-right) feature norm histogram.



(a) SIGReg (tokens). (b) HamJEPA.

Figure 4: q/p **decomposition across pretraining (ImageNet-100).** We evaluate frozen features extracted at sparse pretraining checkpoints using (left) a linear probe top-1 and (right) the best kNN top-1 across the evaluated k values. For SIGReg+tokens, concatenating (q, p) is consistently better than either half alone, indicating that the split halves are complementary (and/or that doubling feature dimension helps). For HamJEPA, q, p, and (q, p) track each other closely, suggesting that both phase-space coordinates are individually informative and largely redundant under the symplectic objective. (Left) SIGReg tokens. (Right) HamJEPA.

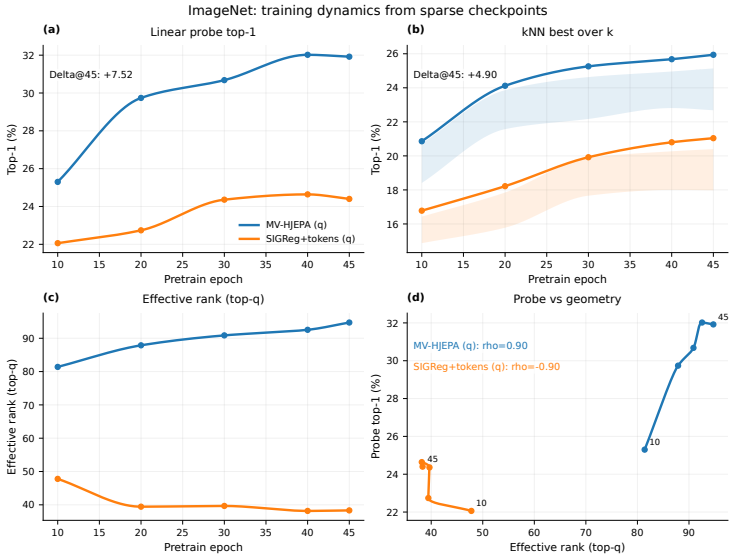


Figure 5: **ImageNet-100 training dynamics from sparse checkpoints (geometry vs. accuracy).** Using the q representation at a set of pretraining epochs, we plot: (a) linear probe top-1; (b) best kNN top-1 over k (shaded region indicates the spread over the k sweep); (c) effective-rank proxy computed from the top-256 eigenvalues of the mean-centered q covariance; (d) probe accuracy versus effective rank (points annotated by epoch), with the plotted correlation coefficient ρ. HamJEPA improves steadily and shows increasing effective rank, while SIGReg+tokens improves modestly while its effective rank decreases; the deltas at epoch 45 are annotated in panels (a,b).

F.4 Replicated controls, metric robustness, and sensitivity

Protocol and independent repetitions. The replicated CIFAR-100 comparison uses two independent 80-epoch pretraining seeds (42 and 43) under the controlled headless token protocol. Each linear score averages five probe optimizations on each frozen representation; uncertainty across pretraining seeds is distinct from probe optimization variability. The q readout has 1024 dimensions for every method. For SIGReg this is the first representation block, not a position coordinate. SIGReg-full uses all 2048 dimensions. The original three-seed 30-epoch experiment and original single-run 80-epoch experiment remain separate cohorts. All results below are descriptive; two pretraining seeds do not support a claim of formal significance.

Evaluation definitions for the new CIFAR-100 experiments. The replicated comparisons and sensitivity study use cached pre-projector features from the frozen encoder in evaluation mode. Evaluation uses all 50,000 training images and 10,000 test images, with tensor conversion and channel normalization using mean $(0.5071, 0.4867, 0.4408)$ and standard deviation $(0.2675, 0.2565, 0.2761)$, without evaluation augmentation. The stored feature layout defines q as the first 1024 coordinates, p as the last 1024, and qp as all 2048.

All four neighborhood evaluations use uniform class voting at $k \in \{1, 5, 10, 20, 50, 100, 200\}$, without temperature weighting. Class-vote ties select the lowest class index. Equal-distance neighbor ordering follows `torch.topk` and is not guaranteed to be identical across hardware. For a feature vector x, define $\nu(x) = x/\max(\|x\|_2, 10^{-12})$. The four modes are:

- **Raw cosine:** compare $\nu(x)$ by dot product.
- **Centered cosine:** subtract the training-feature mean μ from both training and test features, then compare $\nu(x - \mu)$ by dot product.
- **Euclidean:** rank neighbors by squared Euclidean distance on the original features, without centering, whitening, LayerNorm, or row normalization.
- **Whitened cosine:** subtract the same training mean, apply the whitening transform defined below, then normalize each transformed feature before computing dot products.

For whitening, let $N_{\mathrm{tr}} = 50{,}000$ and let d denote the selected readout dimension. Fit only on training features:

$$\mu = \frac{1}{N_{\mathrm{tr}}} \sum_{i=1}^{N_{\mathrm{tr}}} x_i, \qquad \Sigma = \frac{1}{N_{\mathrm{tr}} - 1} \sum_{i=1}^{N_{\mathrm{tr}}} (x_i - \mu)(x_i - \mu)^\top,$$

$$\rho = 10^{-3} \frac{\mathrm{tr}(\Sigma)}{d}, \quad LL^\top = \Sigma + \rho I.$$

The whitened feature is $\nu(L^{-1}(x - \mu))$; for row-stacked features, the corresponding transform is $X_c L^{-\top}$ before row normalization. The implemented dense path forms and symmetrizes the covariance in fp32, performs Cholesky factorization in fp64, and casts the factor to fp32 for the feature transform. Every recovered production result uses the stated ridge without adjustment. The ridge is fixed by this formula; test features and labels do not fit or tune preprocessing.

Probe optimization and uncertainty. For the replicated core and control runs, the probe is a learned LayerNorm with affine scale and bias and $\varepsilon = 10^{-5}$, followed by a linear classifier with 100 outputs. It is trained on the original cached features, without the kNN centering or whitening transforms, using cross-entropy and AdamW for 50 epochs, batch size 256, learning rate 10^{-3}, weight decay 10^{-6}, shuffled training batches, and cosine learning-rate decay at each optimization step. The reported accuracy is from the final probe epoch. Probe seeds are $0, 1, 2, 3, 4$ for each checkpoint.

Each checkpoint's five probe accuracies are averaged first. Reported uncertainty is then the sample standard deviation across the two independent pretraining seeds, using denominator $n - 1$ with $n = 2$. The ten probe optimizations are not ten independent pretraining replications. For these tables, means, sample standard deviations, and improvements are computed from saved values before rounding; each reported statistic is rounded once to two decimal places using round-half-up.

Full-spectrum covariance diagnostics. The new effective-rank and participation-ratio results use all training features of the selected readout, centered by their training mean, without LayerNorm, row

normalization, whitening, or ridge. If s_j are all singular values of the centered feature matrix, define $\lambda_j = s_j^2/(N_{\mathrm{tr}} - 1)$ and retain positive eigenvalues. With $w_j = \lambda_j / \sum_\ell \lambda_\ell$,

$$r_{\mathrm{eff}} = \exp\left(- \sum_j w_j \log w_j \right), \qquad \mathrm{PR} = \frac{\left(\sum_j \lambda_j \right)^2}{\sum_j \lambda_j^2}.$$

An empty positive spectrum gives zero for both statistics. These full-spectrum results are distinct from the historical top-256 PCA estimates computed on at most 5,000 samples.

F.4.1 Replicated performance and the SIGReg readout correction

Method / readout	Raw	Centered	Euclidean	Whitened	Linear
SIGReg-q	25.61	25.62	27.45	27.68	32.22
SIGReg-full	27.50	27.53	29.23	29.63	34.21
MLP-HJEPA-q	27.50	27.48	27.66	36.12	41.47
VPR-NS-q	34.59	34.33	33.97	39.01	44.18
Non-reversible VPR-NS-q	34.93	34.89	33.90	39.18	44.22
HamJEPA-q	34.74	34.59	34.37	39.26	43.88

Table 5: **Replicated CIFAR-100 accuracy after 80 epochs.** The first four columns are kNN@20 with raw cosine, centered cosine, Euclidean, and whitened evaluation. Values are means in percent over two pretraining seeds; linear scores average five probes per seed.

Method / readout	Raw	Centered	Euclidean	Whitened	Linear
SIGReg-q	0.11	0.35	0.03	0.13	0.23
SIGReg-full	0.66	0.81	0.44	0.31	0.55
MLP-HJEPA-q	0.57	0.48	0.73	0.50	0.36
VPR-NS-q	0.11	0.37	0.05	0.29	0.03
Non-reversible VPR-NS-q	0.11	0.15	0.08	0.49	0.17
HamJEPA-q	0.28	0.65	0.30	0.35	0.15

Table 6: **Sample standard deviations across the two pretraining seeds.** Values are in percentage points, with corresponding means in Table 5. Each linear observation is the mean of five probes for one frozen representation.

The original SIGReg values 27.98 raw kNN@20 and 33.95 linear used the full 2048-dimensional representation. The 25.61 and 32.22 values use its 1024-dimensional first block. Fresh full-state runs give 27.97/34.60 (raw kNN@20/linear) for training seed 42 and 27.03/33.82 for seed 43, with means 27.50 ± 0.66 and 34.21 ± 0.55. Thus the first-half and full-state comparisons must not be interchanged. Relative to the fresh full-state SIGReg means, HamJEPA-q gains 7.24 raw kNN@20 and 9.67 linear points. Its centered, Euclidean, and whitened gains over the full representation are respectively 7.07, 5.14, and 9.63 points.

The repeated original value 33.95 has independent origins: the three HamJEPA 30-epoch probe values 34.13, 33.68, and 34.03 average to 33.9467, while SIGReg's 80-epoch result came from a separate stored evaluation. Five fresh probes of the original SIGReg representation give 33.90 ± 0.07; this is probe optimization variability, not a five-seed pretraining result.

Metric and mean-direction checks. HamJEPA exceeds dimension-matched SIGReg and the residual MLP in all 56 q-readout comparisons across four neighborhood geometries, seven values of k, and two pretraining seeds. These correlated sweep points are robustness checks, not independent significance tests. Centering retains approximately 98.3% of the raw HamJEPA–SIGReg gap. The HamJEPA random-pair cosine mean falls from approximately 0.62 to approximately 0.002 after centering, while centered kNN@20 remains 34.59%. The seedwise centered gaps over SIGReg-q are 8.27 and 9.68 points; the corresponding Euclidean gaps are 6.73 and 7.11. The improvement therefore survives removal of the common mean direction and the evaluated changes of metric, and the 1024-dimensional q representation also remains ahead of full-dimensional SIGReg.

F.4.2 Matched non-symplectic and non-reversible controls

Reversible force-rotation control. VPR-NS uses the identical separable potential and 328,449 predictor parameters as HamJEPA, with the same $K = 2$, $h = 0.05$, residual schedule, matching loss, anti-collapse terms, optimizer, batch size, precision, and training schedule. It replaces each potential-force vector $\nabla V_\phi(q)$ by $R\nabla V_\phi(q)$, where R applies a fixed 30-degree rotation to adjacent coordinate pairs, chosen before downstream evaluation. This preserves force norms and gives force alignment $\cos(\pi/6) \approx 0.866$. The kick–drift–kick map remains invertible, reversible, and determinant-one, but is generically non-symplectic; its measured relative symplectic defect is approximately 0.01, versus numerical zero for HamJEPA.

HamJEPA–VPR-NS centered-kNN differences are -0.46 and $+0.98$ points across the two seeds. The linear means are 43.88 and 44.18. These results do not establish a HamJEPA advantage over this control or identify symplecticity as necessary for the benchmark gain.

Non-reversible splitting control. Starting from VPR-NS, the later control changes the palindromic second-order splitting to a predeclared non-palindromic second-order splitting. Architecture, force, horizon, number of force evaluations, and training settings are unchanged. Define the signed macro-step $\tau = \text{direction} \times Kh$, with $K = 2$ and $h = 0.05$, and the stages

$$\mathsf{K}_b^\tau(q,p) = \big(q, p - b\tau R\nabla V_\phi(q)\big), \qquad \mathsf{D}_a^\tau(q,p) = \big(q + a\tau p, p\big).$$

The implemented application order, first applied first, is

$$\mathsf{K}_{0.20}^\tau, \quad \mathsf{D}_{0.40}^\tau, \quad \mathsf{K}_{0.25}^\tau, \quad \mathsf{K}_{0.25}^\tau, \quad \mathsf{D}_{0.60}^\tau, \quad \mathsf{K}_{0.30}^\tau.$$

The two middle kicks are computed separately, giving four potential-gradient evaluations and two drifts, as in the reversible $K = 2$ control. Combining the middle kicks mathematically gives $(a_1, a_2) = (0.4, 0.6)$ and $(b_1, b_2, b_3) = (0.2, 0.5, 0.3)$, which satisfy $a_1 + a_2 = b_1 + b_2 + b_3 = 1$ and $b_1 + a_2 b_2 = 1/2$, the second-order consistency condition. The fixed rotation acts on each adjacent pair as

$$\begin{pmatrix} u_{2j-1} \\ u_{2j} \end{pmatrix} \longmapsto \begin{pmatrix} \cos(\pi/6) & -\sin(\pi/6) \\ \sin(\pi/6) & \cos(\pi/6) \end{pmatrix} \begin{pmatrix} u_{2j-1} \\ u_{2j} \end{pmatrix}, \qquad j = 1, \ldots, 512.$$

Negative direction keeps this application order and negates τ. The exact inverse instead reverses the stage order and negates τ.

The map remains invertible, volume-preserving, generically nonsymplectic, and second-order accurate. Its displacement is matched in the structural fixture. In the eight-dimensional float64 structural test, its determinant error is 2.22×10^{-16}. The unnormalized negative-step cycle norm over a 13×8 fixture batch increases from numerical precision for reversible VPR-NS to 1.17×10^{-7}. The trained-checkpoint reversibility defects, normalized by forward displacement as defined in Appendix F.4.4, are 0.0439 and 0.0165. The corresponding state-normalized cycle errors are approximately 0.00356 and 0.00134; the exact reverse-stage inverse has mean state-normalized errors 2.19×10^{-8} and 2.05×10^{-8}. Its 34.89 ± 0.15 centered q-kNN@20 and 44.22 ± 0.17 linear performance retain the gain. Thus reversibility is not identified as individually necessary either. The experiments do not separately identify volume preservation as necessary.

Scope of the component evidence. The tested constrained phase-space updates outperform SIGReg and the generic residual MLP in this protocol. The auxiliary-matching intervention below supports a role for coupling p during training, while the MLP shows that this supervision alone is insufficient for the best native neighborhoods. This does not isolate one uniquely causal invariant. The oracle characterization selects the Hamiltonian lift under its stated joint assumptions; it does not assert that every violation of an assumption must reduce every downstream score.

F.4.3 Rollout, potential, and matching sensitivity

Nine predeclared conditions use pretraining seed 42 and the first 30 epochs of the same 80-epoch schedule. They are not continuations or replications of the separate original 30-epoch schedule. The default uses $K = 2$, $h = 0.05$, residual scale 0.02, and $q + 0.2p$ matching. Each condition uses one 50-epoch probe with optimization seed 0. No uncertainty across pretraining seeds is estimated for this single-seed sensitivity study.

Setting	Centered kNN@20	Whitened kNN@20	Linear
$K = 1,\ h = 0.10,\ Kh = 0.10$	32.56	34.71	38.91
$K = 2,\ h = 0.05,\ Kh = 0.10$ (default)	32.33	33.78	38.09
$K = 4,\ h = 0.025,\ Kh = 0.10$	32.67	34.26	37.72
$K = 2,\ h = 0.025$	32.31	34.10	37.10
$K = 2,\ h = 0.10$	31.51	33.01	36.46
Residual scale 0	32.01	33.79	38.05
Residual scale 0.05	33.47	34.18	37.76
Pure q matching (p weight 0)	30.01	30.68	34.55
Full-state matching	32.78	34.71	38.28

Table 7: **Single-seed sensitivity at epoch 30 of a common 80-epoch schedule.** Accuracy in percent. The first three rows hold the rollout horizon fixed; the next two also change the horizon.

At fixed horizon, $K = 1, 2, 4$ span 0.34 centered-kNN points. Halving the default step at fixed K changes centered kNN by -0.02; doubling it also doubles the horizon and weakens all three metrics. Setting the learned residual to zero leaves a linear-symplectic quadratic predictor within 0.32 centered-kNN and 0.04 linear points of default. Increasing residual scale improves centered kNN while slightly reducing the probe, so its effect is not uniform across metrics.

Setting only `p_weight=0` removes auxiliary momentum matching and loses 2.32 centered-kNN and 3.54 linear points. Full-state matching remains close to default. The reported covariance diagnostics move in the opposite direction under pure q matching: effective rank rises from 248.71 to 306.23 and participation ratio from 77.48 to 93.83. Thus larger marginal rank is insufficient to explain the accuracy improvement. These are single-seed component trends; hidden width/depth and every anti-collapse threshold were not varied.

F.4.4 Distance from identity and learned dynamics

Diagnostic definitions. For each of $N = 5{,}000$ paired augmentations, let z_i be the initial state, y_i its paired-view target, $\widehat{z}_i = F_+(z_i)$ the prediction, and $c_i = F_-(\widehat{z}_i)$ the cycle obtained using the same stage order with negative direction. With $\varepsilon = 10^{-12}$, the mean relative displacement and state-normalized cycle error are

$$D_{\mathrm{rel}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\|\widehat{z}_i - z_i\|_2}{\max(\|z_i\|_2, \varepsilon)}, \qquad C_{\mathrm{state}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\|c_i - z_i\|_2}{\max(\|z_i\|_2, \varepsilon)}.$$

The reported trained-checkpoint non-reversibility defect instead normalizes each cycle error by its forward displacement:

$$C_{\mathrm{increment}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\|c_i - z_i\|_2}{\|\widehat{z}_i - z_i\|_2 + \varepsilon}.$$

Exact-inverse cycle errors use $F_+^{-1}(\widehat{z}_i)$ in place of c_i in C_{state}. Full-state diagnostics use all 2048 coordinates; block diagnostics apply the same formulas to the relevant 1024 coordinates.

Identity and learned prediction MSE are first averaged over coordinates and then over pairs. Their fractional improvement is $(\overline{\mathrm{MSE}}_{\mathrm{id}} - \overline{\mathrm{MSE}}_{\mathrm{pred}})/\max(\overline{\mathrm{MSE}}_{\mathrm{id}}, \varepsilon)$. Increment alignment is the cosine between $\widehat{z}_i - z_i$ and $y_i - z_i$, using $\varepsilon = 10^{-12}$.

For $\mathcal{H}_\phi(q, p) = \frac{1}{2}\|p\|_2^2 + V_\phi(q)$, the reported mean absolute relative energy drift is

$$\frac{1}{N}\sum_{i=1}^{N} \frac{|\mathcal{H}_\phi(\widehat{z}_i) - \mathcal{H}_\phi(z_i)|}{\max(|\mathcal{H}_\phi(z_i)|, \varepsilon)}.$$

Energy uses coordinate sums, and this relative statistic depends on the energy gauge. The residual/base-force ratio averages $\|s\nabla f_\phi(q_i)\|_2/\max(\|\alpha q_i\|_2, \varepsilon)$ over pairs, where $V_\phi(q) = \frac{1}{2}\alpha\|q\|_2^2 + s f_\phi(q)$. Diagnostic distributional standard deviations use denominator N; the accuracy uncertainties across training seeds instead use sample standard deviations.

The separate float64 structural tests use the full Jacobian J_F in a small-dimensional fixture. With $\Omega = \left(\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right)$, the relative symplectic defect is $\|J_F^\top \Omega J_F - \Omega\|_F / \|\Omega\|_F$, and the determinant error is

$|\det J_F - 1|$. These fixture measurements are not full-Jacobian tests of the trained 2048-dimensional predictors.

Diagnostics on 5,000 paired augmentations per checkpoint give approximately 8.3% full-state displacement for HamJEPA and 8.0% for reversible VPR-NS. Both use p functionally and invert with cycle error approximately 2×10^{-8}. HamJEPA's mean absolute energy drift is 0.145%, compared with 0.488% for VPR-NS, and its learned residual force is roughly five times the quadratic force. Thus the trained predictor is neither the inactive identity nor merely its fixed quadratic base.

HamJEPA nevertheless does not reduce final paired-view MSE relative to identity. The residual MLP improves that MSE by approximately 29% while remaining 7.11 centered-kNN and 2.41 linear points below HamJEPA. Zeroing p improves HamJEPA/VPR q-prediction MSE, whereas shuffling or randomizing p changes the prediction substantially. These diagnostics distinguish representation quality from direct paired-view regression fit. They support interpreting the predictor as a geometric training constraint and do not establish a physical trajectory that transports each augmentation pointwise to the other.

F.4.5 Matched training cost

Method	Seconds / step	Augmented views / s	Peak allocated (GiB)
SIGReg	0.664	1542	2.457
MLP-HJEPA	0.808	1268	2.459
VPR-NS	0.838	1222	2.459
HamJEPA	1.696	604	2.459

Table 8: **Matched exact-configuration A100 preflight benchmark on real CIFAR-100 batches.** Batch size is 512 images with two views per image; the throughput column counts augmented views. Each entry averages two cold preflight observations, using zero warmup steps and one timed step per method and seed configuration. These preflight measurements are distinct from the approximate original full-run timings in Appendix H.1.

For HamJEPA, the cold preflight step time is approximately $2.55\times$ that of SIGReg, with essentially unchanged peak allocated memory; the zero-warmup, single-step procedure does not support a reliable steady-state runtime ratio. It adds 328,449 predictor parameters. The predictor is used only during pretraining and is discarded for frozen-encoder inference.

The benchmark reconstructs newly initialized temporary model and optimizer instances and uses a fixed real augmented batch transferred to the GPU before timing. Data loading and augmentation preparation are excluded. The timed full step includes the encoder/projector, objective and active regularizers, backward pass, configured gradient clipping, optimizer, and scheduler. Timing uses CUDA synchronization before and after each step, with peak allocated-memory statistics reset before timing.

Remaining evaluation scope. ImageNet-100 remains a single 45-epoch pretraining seed. Production learned-marginal HamSIGReg, standard projector/EMA/multicrop recipes, VICReg/Barlow Twins/KerJEPA, larger backbones, and ImageNet-1k were not completed. The controlled results do not establish that the gain survives those settings or that the theoretical price of isotropy equals the measured accuracy difference.

F.5 Further figures

Figures 6–8 provide an intuition for a LeJEPA-inspired *sliced-projection* diagnostic: rather than comparing rollout distributions only in aggregate, we compare many 1D projections (a Cramér–Wold viewpoint) to expose direction-dependent geometric distortion.

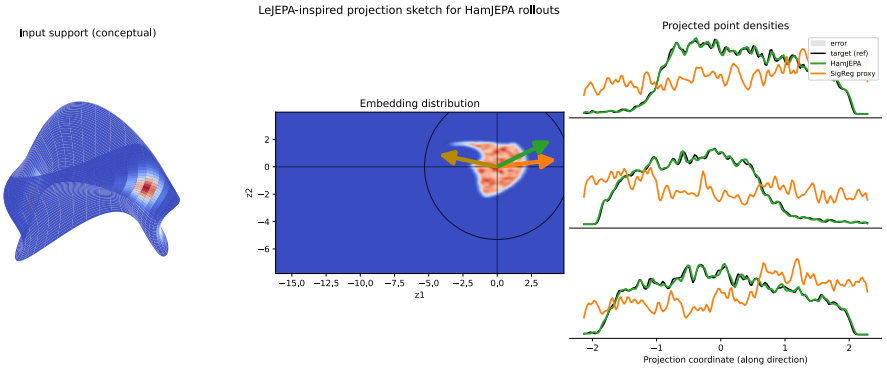


Figure 6: **LeJEPA-inspired sliced-projection diagnostic for HamJEPA rollouts (cartoon).** Left: conceptual support of the input distribution (manifold-structured data). Middle: an example embedding distribution in a 2D slice with several projection directions $u(\theta)$. Right: estimated 1D projected densities along representative directions; black denotes the reference/target, green denotes HamJEPA rollouts, orange denotes a SIGReg proxy baseline, and the shaded band visualizes the per-direction projection error. Matching many 1D projections (Cramér–Wold/sliced view) makes geometric distortion from non-structure-preserving dynamics visible, while symplectic rollouts remain close to the reference across directions.

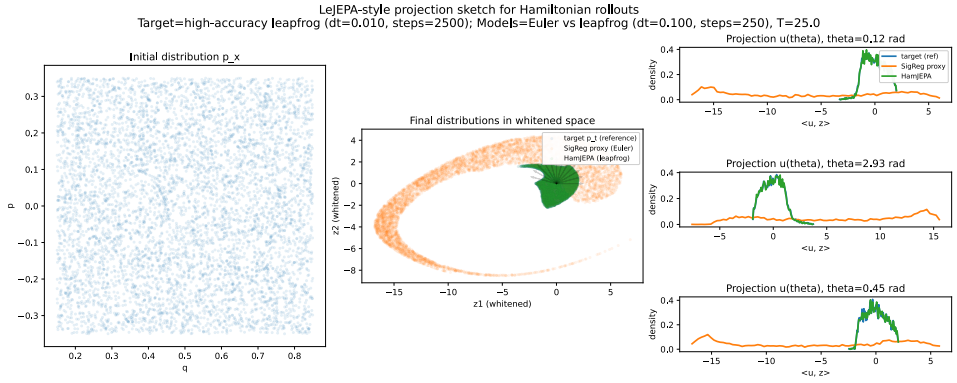


Figure 7: **LeJEPA-style projection sketch on a toy Hamiltonian transport problem.** Left: initial phase-space distribution p_x over (q, p). Middle: final distributions in whitened coordinates after rolling out to a fixed horizon T; the reference target is generated by a high-accuracy leapfrog integrator, while the two model curves compare a coarse Euler rollout (SIGReg proxy) against a coarse leapfrog rollout (HamJEPA). Right: 1D densities of projections $\langle u(\theta), z \rangle$ for representative angles θ, showing that leapfrog-based rollouts track the reference distribution across directions, whereas Euler produces large direction-dependent mismatch.

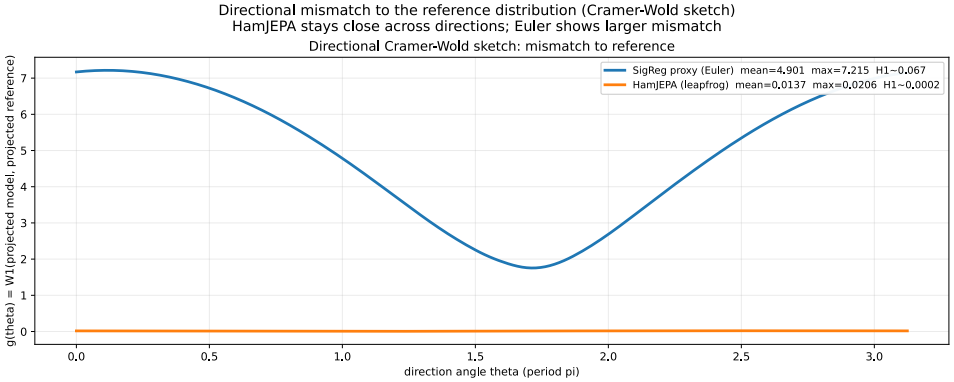


Figure 8: **Directional Cramér–Wold discrepancy (sliced mismatch) versus projection angle.** For each direction angle θ (period π), we compute a 1D mismatch $g(\theta) = W_1(\langle u(\theta), Z_{\text{model}}\rangle, \langle u(\theta), Z_{\text{ref}}\rangle)$ between the projected model and reference distributions. HamJEPA (leapfrog) stays uniformly close across directions, while the Euler/SIGReg proxy exhibits large, anisotropic deviation; the legend reports summary statistics over θ (mean/max and the aggregate score used in the plot).

G A visual guide to HamJEPA

This appendix gives a diagrammatic explanation of HamJEPA for readers who want a more visual account than the equation-level presentation in the main text. The figures here are explanatory schematics rather than empirical results. Their purpose is to separate the method into four conceptual layers: (i) the overall two-view prediction pipeline, (ii) how the encoder constructs the phase-space state $s = [q; p]$, (iii) how the predictor evolves that state by a symplectic Hamiltonian rollout, and (iv) why encoder-side anti-collapse regularization is still required even when the predictor itself is symplectic.

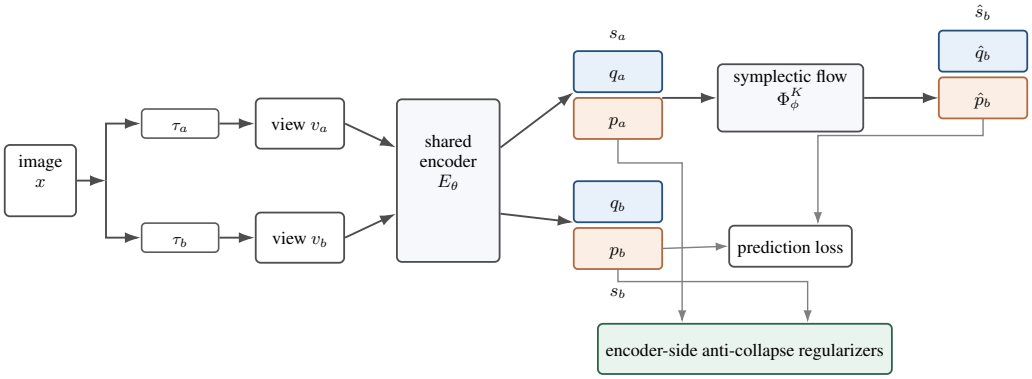


Figure 9: **HamJEPA at the highest level.** Two augmented views of the same image are encoded by a shared encoder into phase-space states $s_a = [q_a; p_a]$ and $s_b = [q_b; p_b]$. The predictor is not an unconstrained regression head: it is a structured Hamiltonian rollout Φ_ϕ^K that evolves s_a into a predicted target state $\hat{s}_b$. Training combines a prediction loss between $\hat{s}_b$ and s_b with separate encoder-side anti-collapse regularizers applied across the batch of states.

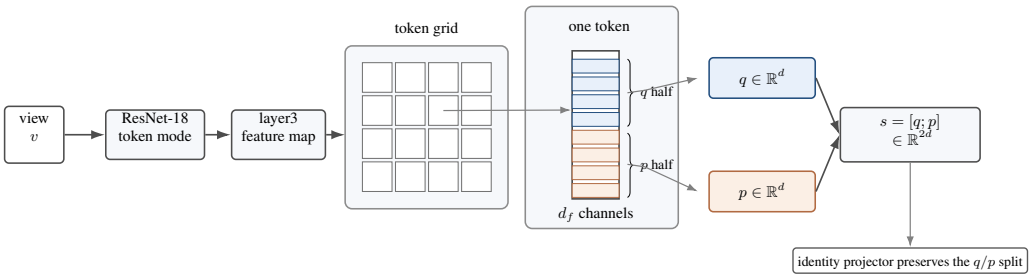


Figure 10: **How the encoder constructs the phase-space state.** HamJEPA is not just taking an arbitrary vector and renaming its halves as q and p. In the implementation, the encoder runs in token mode: an intermediate ResNet feature map is converted into a token grid, the *channel stack of each token* is split into a q half and a p half, and those halves are flattened across the spatial grid to form $q \in \mathbb{R}^d$ and $p \in \mathbb{R}^d$. These are then concatenated into $s = [q; p] \in \mathbb{R}^{2d}$. This is why HamJEPA uses an identity projector: an unconstrained MLP after the encoder could mix the two coordinate blocks.

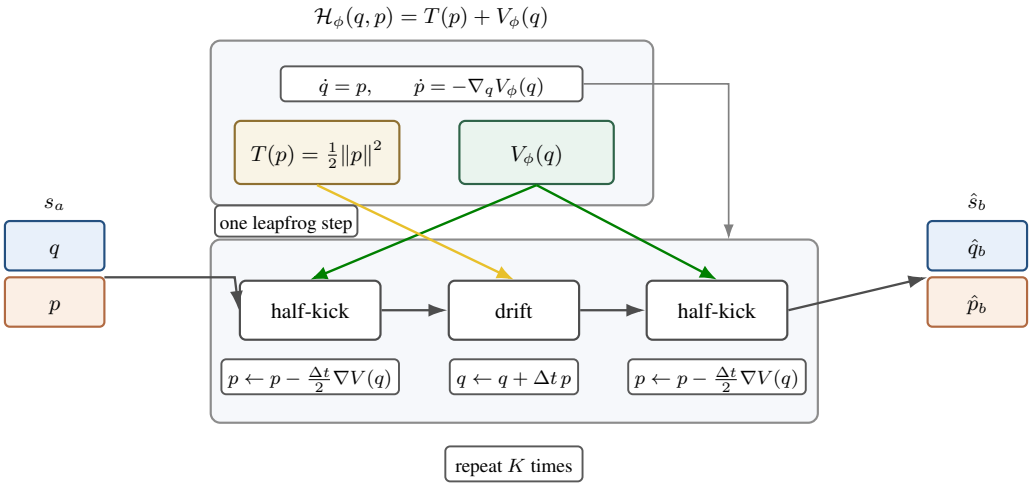


Figure 11: **What the predictor is doing.** HamJEPA does not use an arbitrary predictor $z_a \mapsto \hat{z}_b$. Instead, it rolls the state forward with a learned separable Hamiltonian $\mathcal{H}_\phi(q,p) = T(p) + V_\phi(q)$ and a leapfrog integrator. The potential $V_\phi(q)$ drives the kick updates through $\nabla_q V_\phi(q)$, while the kinetic term $T(p) = \frac{1}{2}\|p\|^2$ yields the drift update $q \leftarrow q + \Delta t\, p$. Repeating this structured step K times produces the predicted target state $\hat{s}_b = \Phi_\phi^K(s_a)$.

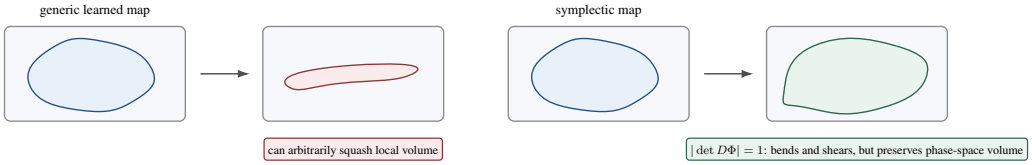


Figure 12: **What "symplectic" means geometrically.** A generic learned map can bend, stretch, and arbitrarily collapse local geometry. A symplectic map can still deform the state space, but it preserves total phase-space volume locally, i.e. $|\det D\Phi| = 1$. This gives local invertibility and ensures that the predictor itself is not the source of volume collapse.

Symplecticity constrains the predictor, but it does *not* by itself prevent encoder collapse. A collapsed encoder could still map many different views to almost the same latent state, and a perfectly symplectic predictor would then only move around an already-degenerate state cloud. The next two figures therefore separate these two roles: first, why predictor structure alone is insufficient; second, how the encoder-side regularizers keep the latent space alive.

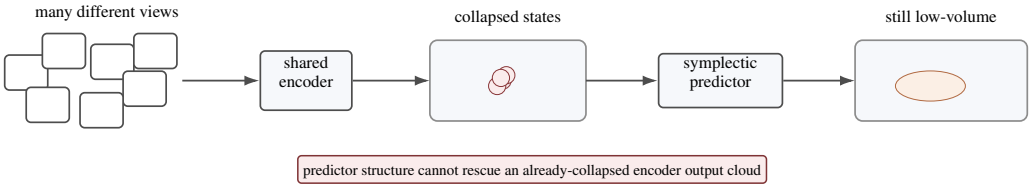


Figure 13: **Why the predictor alone is not enough.** Even if the predictor is perfectly symplectic, the encoder could still map many different views to nearly the same latent state. In that situation the rollout acts on an already-collapsed representation. This is why HamJEPA needs a second component beyond the predictor: encoder-side anti-collapse regularization.

Why the predictor alone is not enough. The symplectic constraint belongs to the predictor map, not to the encoder. This means it prevents the rollout from crushing phase-space volume, but it does *not* stop the encoder itself from mapping many different views to nearly the same latent state. A

collapsed encoder would therefore still give a degenerate state cloud, even with a perfectly symplectic predictor. The next figures isolate the role of each encoder-side anti-collapse regularizer.

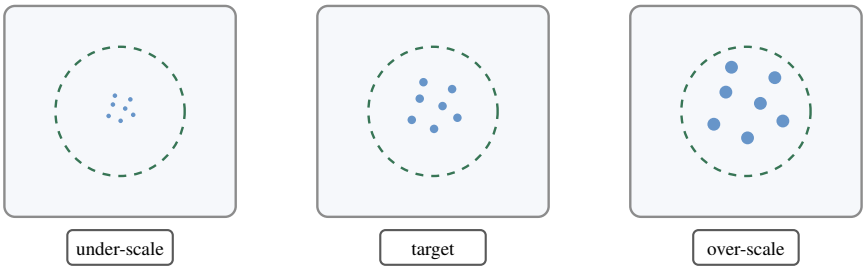


Figure 14: **Energy budget.** The energy budget keeps the average per-dimension scale of q and p near a target range. It penalizes both under-scaled states (which risk collapse toward zero) and over-scaled states (which can destabilize dynamics), so the latent cloud stays at a controlled overall radius.

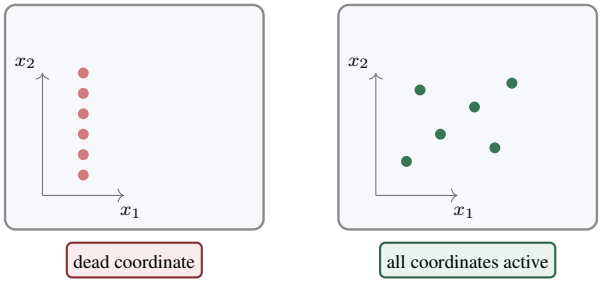


Figure 15: **Variance floor.** When satisfied, the variance floor prevents individual coordinates from becoming nearly constant across the batch. Visually, it revives dead axes: each coordinate retains a minimum amount of variation. This does not by itself certify full-rank covariance.

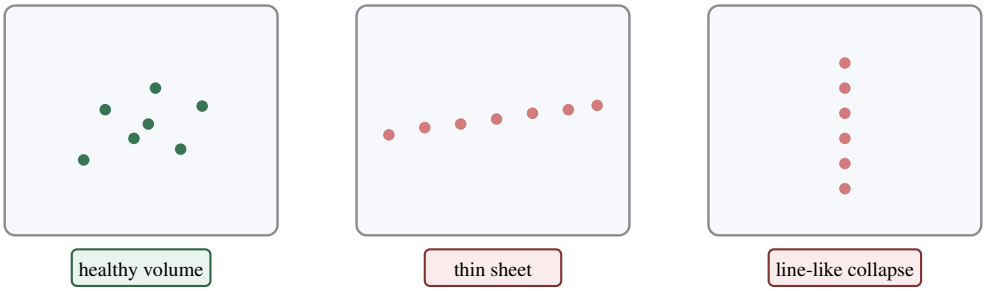


Figure 16: **Projected log-det floor.** When satisfied, the projected log-det floor bounds the determinant of the regularized projected covariance away from zero. It does not require isotropy, but it does force that covariance to retain nontrivial volume. Unregularized rank protection additionally requires a trace bound and a lower eigenvalue bound exceeding the covariance ridge (Proposition B.13).

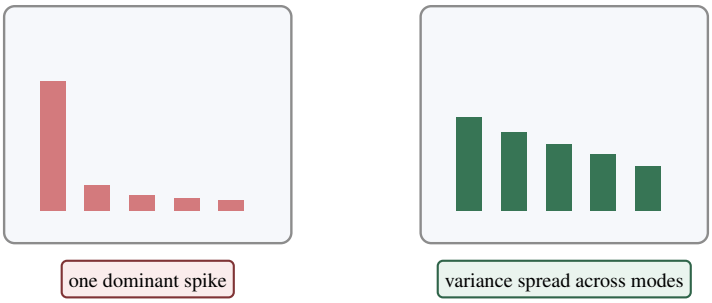


Figure 17: **Participation-ratio floor and top-eigenvalue control.** These penalties discourage the representation from surviving only through one huge dominant direction. The participation-ratio floor encourages variance to remain spread across multiple modes, while the top-eigenvalue penalty discourages a single spike from taking over the spectrum.

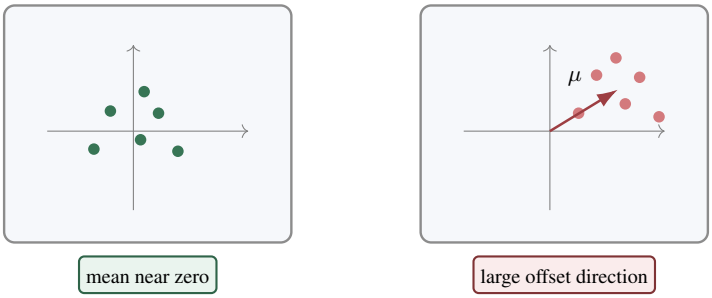


Figure 18: **Mean penalty.** A large global mean can create a dominant offset direction that makes random-pair cosine similarities look artificially cone-like. The mean penalty discourages this by keeping the batch mean closer to zero, so the geometry is not dominated by a single global shift.

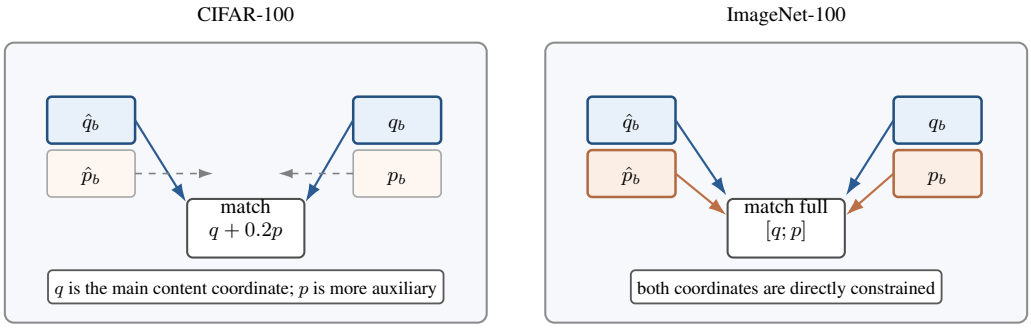


Figure 19: **Why CIFAR-100 and ImageNet-100 use different matching choices.** On CIFAR-100 the training objective emphasizes the content coordinate q with an auxiliary $0.2p$ matching term; q is the cleanest downstream readout and p mainly supports the dynamics. On ImageNet-100 we match the full state $[q; p]$, which directly constrains both coordinates, and p is much more informative.

How to read the sequence. Taken together, Figs. 9–19 decompose the full method into three conceptual layers: the encoder constructs a structured phase-space state, the predictor evolves that state with a symplectic Hamiltonian rollout, and the regularizers ensure that the encoder supplies a non-degenerate cloud of states for those dynamics to act on. The distinctive point is therefore twofold: HamJEPA changes both the *representation parameterization* (from a single embedding to a phase-space state) and the *predictor family* (from arbitrary regression to Hamiltonian dynamics), while avoiding the need to force the encoder marginal toward Euclidean isotropy.

H Implementation Details and Algorithms

In this section (as in the code) we refer to *HamJEPA* as *MV-HJEPA*, denoting the minimal multi-view implementation used in our experiments.

Source provenance and archive scope. The public code repository is linked in the introduction. The replicated CIFAR-100 experiments, sensitivity study, and common evaluator use source revision `894311e0126ac9786267ed42796640718bfd6e91`. The non-reversible control and additional cycle diagnostics use `4af1d4548153d2724f0854514c555f6f64d22b3e`. The accompanying supplementary reproducibility archive records these source snapshots, configurations, persisted execution commands, saved evaluation outputs, file hashes, and external runtime helpers verified against recorded hashes. The archive excludes datasets, checkpoints, and feature tensors; checkpoint hashes are retained. These archived revisions identify the scientific source for the new experiments.

The recovered historical snapshot `5089d6e01902c3a843de24778ae6fb69da929086` supports the documented ImageNet-100 $[+1, +1]$ view directions, full-state matching, and fixed residual scale 0.02, but is not a certified producing revision: the inspected checkpoint contains no source-revision metadata. Its exact source-to-checkpoint provenance remains unresolved.

Asset provenance and terms. CIFAR-100 was created by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton and is described in Krizhevsky [13]. Its official distribution page, `https://www.cs.toronto.edu/~kriz/cifar.html`, requests citation of that report but does not state a separate named dataset license. ImageNet-100 is a subset of ImageNet [7]; ImageNet access is subject to its non-commercial research and educational terms at `https://www.image-net.org/download.php`. The implementation uses PyTorch [16] and torchvision; their BSD 3-Clause terms and copyright notices are provided at `https://github.com/pytorch/pytorch/blob/main/LICENSE` and `https://github.com/pytorch/vision/blob/main/LICENSE`. The SIGReg implementation includes code adapted from LeJEPA [3], with Creative Commons Attribution–NonCommercial 4.0 International (CC BY-NC 4.0) terms documented in the public repository's license and attribution notice at `https://github.com/R02AJ/HamJEPA-HamSIGReg/blob/main/LICENSE`. No dataset images are redistributed.

H.1 Compute resources

All reported experiments were run on NVIDIA A100 GPUs. CIFAR-100 pretraining took approximately 40 seconds per epoch, corresponding to roughly 20 minutes for a 30-epoch run and roughly 53 minutes for an 80-epoch run. For the historical 45-epoch ImageNet-100 runs, the recorded wall times of the training cells were approximately 15.43 hours for HamJEPA and 5.45 hours for SIGReg. These notebook durations include the full cell execution and do not isolate GPU compute time. Downstream evaluation was also run on A100 GPUs and required under 5 minutes for CIFAR-100 and under 15 minutes for ImageNet-100 per checkpoint/readout evaluation. These timings are approximate wall-clock measurements for the original runs; peak GPU memory was not separately profiled for those runs. The matched cold preflight benchmark in Appendix F.4.5 reports single-step time and peak allocated memory.

H.2 Two training modes: MV-HJEPA vs. baseline (LeJEPA + regularizer)

Our training scripts are *mode-switched* by the presence of a `hjepa:` block in the YAML:

- **MV-HJEPA mode** (`hjepa` present): we treat the representation as a phase-space state $z = [q; p] \in \mathbb{R}^{2d}$ and train a Hamiltonian flow predictor that maps between two global views. We additionally apply scale/diversity regularizers on *all* views.
- **Baseline mode** (`hjepa` absent): we use the *LeJEPA* prediction objective where all views predict the mean of the global views, and we optionally add a distributional regularizer such as SIGReg.

H.3 Data pipeline and multi-view batching (CIFAR100 vs. ImageNet)

Multi-crop dataset contract. Each dataset example yields a list of V augmented views (multi-crop), plus labels:

$$(\{x^{(v)}\}_{v=1}^{V}, y, y_{\text{coarse}}).$$

In our configurations, we set `num_global_views=2` and `num_local_views=0`, so $V = 2$.

CIFAR100MultiCrop. `eval/datasets/cifar100_multicrop.py` applies the full augmentation *in the worker* and returns *float tensors already normalized* (using `ToTensor()` + `Normalize`). Notable defaults:

- `out_size=32` (CIFAR resolution)
- blur uses a small kernel (`kernel_size=3`)

ImageNetMultiCrop. `eval/datasets/imagenet_multicrop.py` applies the same high-level augmentation pattern, but returns *uint8 tensors* using `PILToTensor()` to reduce dataloader IPC bandwidth. We then convert to float and normalize *on device* inside the ImageNet training loop. Notable defaults:

- `out_size=224`
- blur uses a larger kernel (`kernel_size=23`) appropriate for higher resolution
- global crop scale defaults match the YAML (`global_scale=[0.4,1.0]`)

Multi-view batching optimization. For computational efficiency, we concatenate views along the batch dimension and run a *single* encoder+projector forward:

$$x_{\text{cat}} = \text{concat}\left(x^{(1)}, \ldots, x^{(V)}\right) \in \mathbb{R}^{(VB)\times C\times H\times W}, \qquad z_{\text{cat}} = P(E(x_{\text{cat}})) \in \mathbb{R}^{(VB)\times K}.$$

We then reshape:

$$z_{\text{views}} \in \mathbb{R}^{V\times B\times K}, \qquad z_{\text{views}} = \text{reshape}(z_{\text{cat}}, (V, B, K)).$$

This is exactly how both `train_cifar_hamjepa.py` and `train_imagenet_hamjepa.py` operate.

H.4 Encoder and token-state parameterization

ResNet18 backbone with dataset-specific stem. We use `torchvision.models.resnet18(weights=None)` and alter the stem depending on the dataset:

- **CIFAR stem** (`encoder_stem=cifar`): `conv1` is replaced by a 3×3 stride-1 conv and `maxpool` is removed.
- **ImageNet stem** (`encoder_stem=imagenet`): keep the default 7×7 stride-2 conv and maxpool.

Token mode (used in all shown configs). With `encoder_mode=tokens`, we expose an intermediate feature map at a chosen ResNet layer (`token_layer` in {layer2, layer3, layer4}), optionally pool to a fixed spatial grid (`token_hw`), project its channels with a 1×1 conv and batch normalization to `token_d_f`, then flatten.

Concretely, if the token feature map has shape `[B, token_d_f, h, w]`, the flattened representation is

$$z \in \mathbb{R}^{B\times(hw\,\texttt{token_d_f})}.$$

This is why `embed_dim` must match hw `token_d_f`:

- CIFAR100: `layer3` naturally yields an 8×8 map, and with `token_d_f=32` we get $8\cdot8\cdot32 = 2048$.
- ImageNet: `layer3` yields 14×14 pre-pool; we set `token_hw=8` so adaptive pooling yields 8×8, again giving 2048.

The resulting token projection and flattening define the trained encoder output. In the reported runs this output comes from `layer3`; `layer4` is not on the loss path and is not trained as a later version of that representation. Here "headless" refers to the absence of a separate learned projector after this encoder output.

Projector. We support either an MLP projector or an identity projector. In MV-HJEPA we *require* `projector_type=identity` and enforce it in code, because we interpret the representation channels as structured phase-space coordinates and do not want a learned MLP to arbitrarily mix them.

Phase-space split: $z = [q; p]$. In MV-HJEPA we require an even representation dimension $K = 2d$, and interpret:

$$z = \begin{bmatrix} q \\ p \end{bmatrix} \in \mathbb{R}^{2d}, \qquad q \in \mathbb{R}^d,\ p \in \mathbb{R}^d.$$

This split is implemented *channel-wise per token* in the encoder when `split_qp=true`: the token channel dimension `token_d_f` is split into equal halves (first half $\to q$, second half $\to p$) before flattening and concatenation.

H.5 Hamiltonian flow predictor (MV-HJEPA)

Hamiltonian dynamics. We predict one view's state from another by integrating Hamiltonian dynamics in latent phase space. Given a Hamiltonian $H(q, p)$, the vector field is:

$$\frac{dq}{dt} = \frac{\partial H}{\partial p}, \qquad \frac{dp}{dt} = -\frac{\partial H}{\partial q}.$$

In code, $\partial H/\partial q$ and $\partial H/\partial p$ are obtained via automatic differentiation (`torch.autograd.grad(create_graph=True)`), so gradients propagate through the integrator.

Hamiltonian parameterizations. `hamjepa/hamiltonian.py` supports:

1. **Quadratic**: $H(q, p) = \frac{1}{2}(\|q\|^2 + \|p\|^2)$ (no learnable parameters).
2. **Learnable (residual MLP)**:

   $$H_\theta(q, p) = \tfrac{1}{2}(\|q\|^2 + \|p\|^2) + s\, f_\theta([q, p]),$$

   where s (`residual_scale`) is a stabilizing residual multiplier.
3. **Separable (used for MV-HJEPA)**:

   $$H_\phi(q, p) = T(p) + V_\phi(q), \qquad T(p) = \tfrac{1}{2}\|p\|^2, \qquad V_\phi(q) = \tfrac{1}{2}\alpha\|q\|^2 + s\, f_\phi(q),$$

   where α (`base_coeff`) is a fixed quadratic base coefficient and s (`residual_scale`) is a residual scaling factor.

The separable form enables an explicit velocity-Verlet/leapfrog update that is symplectic.

Residual-scale warmup. In CIFAR-100 MV-HJEPA, we linearly warm up `residual_scale` over `residual_scale_warmup_epochs`, setting $s_e = 0.02 \min(1, e/10)$ at epoch $e \geq 1$. Early in training the potential is close to purely quadratic (stable), and the learned residual is introduced gradually. The ImageNet-100 training loop does not apply this residual-scale schedule: its potential uses the fixed multiplier $s = 0.02$ despite the warmup field in the configuration. Both datasets use the separately configured learning-rate warmup.

Step size and fp32 integration. We integrate for `steps=S` steps with step size `dt` (fixed in MV-HJEPA; `learn_dt=false`). For mixed precision runs (bf16), we optionally do the dynamics computations in fp32 and cast back (`integrate_fp32=true`) to stabilize higher-order autodiff through the integrator.

H.6 Symplectic integrators

We implement:

- **Symplectic Euler** for general $H(q, p)$.
- **Leapfrog / velocity-Verlet**:
  - a general variant using autodiff for $\nabla_q H$ and $\nabla_p H$
  - a specialized separable variant (used by MV-HJEPA when $H = T + V$) using only $\nabla V(q)$

For separable $H(q, p) = \frac{1}{2}\|p\|^2 + V(q)$, the specialized update is:

$$p_{1/2} = p - \tfrac{\Delta t}{2}\nabla V(q), \quad q \leftarrow q + \Delta t\, p_{1/2}, \quad p \leftarrow p_{1/2} - \tfrac{\Delta t}{2}\nabla V(q).$$

H.7 MV-HJEPA prediction loss

Given two global views' states $z_a = [q_a; p_a]$ and $z_b = [q_b; p_b]$, we predict:

$$\hat{z}_b = \mathrm{Flow}_H(z_a; \Delta t, S),$$

and compute a MSE matching term either on q with an auxiliary p term or the full (q, p) state:

$$\mathcal{L}_{\text{match}} = \begin{cases} \mathrm{MSE}(\hat{q}_b, q_b) + w_p\, \mathrm{MSE}(\hat{p}_b, p_b) & \text{(match = } q\text{)} \\ \mathrm{MSE}(\hat{z}_b, z_b) & \text{(match = } qp\text{)}. \end{cases}$$

We stop-gradient the target branch (`detach_target=true`); encoder-side regularizers discourage trivial collapse. CIFAR-100 uses $w_p = 0.2$, while ImageNet-100 uses `match=qp`.

Optionally, we add an energy-consistency term that encourages the flow to preserve the input energy:

$$\mathcal{L}_{\text{energy}} = \mathrm{MSE}\big(H(q_a, p_a),\, H(\hat{q}_b, \hat{p}_b)\big),$$

and the full prediction loss is:

$$\mathcal{L}_{\text{pred}} = \mathcal{L}_{\text{match}} + \lambda_E \mathcal{L}_{\text{energy}}.$$

Bidirectional training. If enabled, we symmetrize by predicting in both directions.

$$\mathcal{L}_{\text{pred}} \leftarrow \tfrac{1}{2}\left(\mathcal{L}_{\text{pred}}(z_0, z_1) + \mathcal{L}_{\text{pred}}(z_1, z_0)\right).$$

In the CIFAR-100 implementation, the reverse direction uses reverse-time integration (negated Δt), which is supported by the predictor's `direction` flag. In the ImageNet-100 implementation, the loss module is unidirectional and the loop averages two calls with swapped view inputs; both calls use positive Δt.

H.8 Exact spectral terms in the objective

The following definitions complete the objective in Section 4.3.

(4) Projected log-det volume floor. A scale budget alone can be satisfied by concentrating variance in a low-dimensional subspace. To prevent rank collapse without forcing isotropy, we impose a projected log-det floor on the content batch. Let $Q \in \mathbb{R}^{B \times d_0}$ contain centered content codes: $Q_i = q_i - \frac{1}{B}\sum_{j=1}^{B} q_j$. Sample a random projection $R \in \mathbb{R}^{d_0 \times k}$ with orthonormalized columns ($k \ll d_0$) and form $Y = QR \in \mathbb{R}^{B \times k}$. Define the regularized projected covariance

$$\Sigma_Y := \frac{1}{B-1} Y^\top Y + \varepsilon I_k, \qquad \varepsilon > 0, \tag{47}$$

and per-dimension log-volume

$$\ell_{\text{vol}} := \frac{1}{k} \log\det(\Sigma_Y). \tag{48}$$

We use the hinge penalty

$$\mathcal{L}_{\text{vol}} := \big[\max(0,\ \tau - \ell_{\text{vol}})\big]^2, \qquad \tau \in \mathbb{R}. \tag{49}$$

(5) Participation ratio floor (discourages "one-spike" solutions). Log-det alone can be satisfied with one very large eigenvalue and many tiny ones when scale is not tightly controlled, or when minibatch effects dominate. To penalize the "one-spike" regime, we add a participation ratio (PR) floor on Σ_Y:

$$\mathrm{PR}(\Sigma_Y) := \frac{\big(\mathrm{tr}(\Sigma_Y)\big)^2}{\mathrm{tr}(\Sigma_Y^2)} = \frac{\big(\sum_{i=1}^k \lambda_i\big)^2}{\sum_{i=1}^k \lambda_i^2} \in [1, k], \tag{50}$$

where $\lambda_1, \ldots, \lambda_k$ are the eigenvalues of Σ_Y. We penalize violations of a target effective rank $r_0 \in (1, k]$:

$$\mathcal{L}_{\text{pr}} := \big[\max(0,\ r_0 - \mathrm{PR}(\Sigma_Y))\big]^2. \tag{51}$$

For each block $x \in \{q, p\}$, the implementation uses the normalized PR hinge, $k^{-2}\mathcal{L}_{\text{pr}}^x$, with $r_0^x = k\rho_x$, and adds the top-eigenvalue-fraction penalty

$$\mathcal{L}_{\text{eig}}^x = \left[\max\left(0, \frac{\lambda_{\max}(\Sigma_Y^x)}{\mathrm{tr}(\Sigma_Y^x)} - \gamma_x\right)\right]^2.$$

H.9 Regularizers (applied across all views)

We compute regularizers on the concatenation of all views' states:

$$z_{\text{all}} \in \mathbb{R}^{(VB)\times 2d}.$$

Phase-space energy budget (scale control). We stabilize scale in fixed units by enforcing per-coordinate second moments:

$$\mathcal{L}_{\text{budget}} = \big(\mathbb{E}[q^2] - q_{\text{tgt}}\big)^2 + \big(\mathbb{E}[p^2] - p_{\text{tgt}}\big)^2,$$

where expectations are taken over batch and coordinates in fp32 and can be Distributed Data Parallel (DDP)-synchronized.

Variance floor (anti-collapse). Let σ_i be the batch standard deviation of dimension i (fp32 stats). We apply:

$$\mathcal{L}_{\text{var}} = \frac{1}{d}\sum_{i=1}^{d} \max(0, \sigma_{\min} - \sigma_i)^2,$$

on both q and p for CIFAR-100 and on q only for ImageNet-100. The standard deviation is computed from the population batch variance with additive $\varepsilon = 10^{-4}$ inside the square root.

Projected log-determinant floor (spectral diversity). We enforce a lower bound on the log-determinant of a randomly projected covariance. Given $x \in \mathbb{R}^{B\times d}$, sample an orthonormal projection $R \in \mathbb{R}^{d\times k}$, compute $y = (x - \bar{x})R$, then:

$$C = \mathrm{Cov}(y) + \varepsilon I, \qquad \mathcal{L}_{\log\det} = \max\Big(0, \tau - \tfrac{1}{k}\log\det(C)\Big)^2.$$

We add constraints on the normalized participation ratio PR/k, where $\mathrm{PR} = \mathrm{tr}(C)^2/\mathrm{tr}(C^2)$, and the top-eigenvalue fraction $\lambda_{\max}/\mathrm{tr}(C)$. Each block's spectral bundle is the sum of the log-det hinge, $\max(0, \rho - \mathrm{PR}/k)^2$, and $\max(0, \lambda_{\max}/\mathrm{tr}(C) - \gamma)^2$, with unit internal weights. The scalar `lambda_logdet` multiplies the sum of the q and p bundles.

Mean penalty. We penalize non-zero feature means using the squared mean (fp32):

$$\mathcal{L}_{\text{mean}} = \frac{1}{2d}\|\mathbb{E}[z]\|_2^2.$$

Active configuration values. Both datasets use two leapfrog steps, $\Delta t = 0.05$, quadratic base coefficient 0.1, residual target scale 0.02, and a residual MLP of width 256 and depth 2. Both use $q_{\text{tgt}} = p_{\text{tgt}} = 1$, standard-deviation floor 0.2, projected dimension $k = 128$, covariance ridge 10^{-4}, and projection refresh interval 16.

Setting	CIFAR-100	ImageNet-100
Matching	$q + 0.2p$	full (q, p)
Return-view step sign	$-\Delta t$	$+\Delta t$
Variance blocks	q, p	q
$(\lambda_{\text{budget}}, \lambda_{\text{var}}, \lambda_{\log\det}, \lambda_{\text{mean}})$	$(0.2, 0.5, 0.5, 0.05)$	$(0.2, 1, 1, 0.05)$
q/p log-det floors	$-1.5, -1.5$	$-1.2, -1.2$
q/p normalized PR floors	$0.12, 0.16$	$0.25, 0.25$
q/p eigmax-fraction ceilings	$0.35, 0.30$	$0.10, 0.10$
Energy-consistency weight	0	0

Frozen-feature evaluation. The evaluation scripts extract features with the encoder frozen and in evaluation mode. CIFAR-100 uses tensor conversion and normalization; ImageNet uses resize, center crop, tensor conversion, and normalization. The reported "linear probe" consists of a learned LayerNorm followed by a linear classifier on these frozen features, trained with cross-entropy and AdamW; it is not an affine classifier directly on the raw features. The reference probe defaults are 50 epochs, batch size 256, learning rate 10^{-3}, weight decay 10^{-6}, and cosine learning-rate decay. Raw cosine kNN uses normalized frozen features and uniform voting.

H.10 Training step summary

MV-HJEPA. At each step:

1. Build V views per image, concatenate them, and compute $z_{\text{views}} \in \mathbb{R}^{V \times B \times 2d}$.
2. Select the two global views (z_0, z_1) and compute $\mathcal{L}_{\text{pred}}$ via Hamiltonian flow prediction.
3. Compute $\mathcal{L}_{\text{budget}}$, $\mathcal{L}_{\text{var}}$, $\mathcal{L}_{\log\det}$, and $\mathcal{L}_{\text{mean}}$ on all views.
4. Optimize encoder + (identity) projector + predictor parameters with AdamW, optional gradient clipping, and cosine LR schedule with warmup.

Baseline (LeJEPA + SIGReg/HamSIGReg). If `hjepa` is absent:

1. Compute z_{views} as above.
2. Apply LeJEPA prediction loss: all views match the per-sample mean of global views.
3. Add a distributional regularizer (e.g. SIGReg) with weight `lambda_reg`.
4. If the regularizer includes a learnable metric H, optionally update H on a slower/periodic schedule.

H.11 Algorithms

Notation. Let V be the number of augmented views per image, B the batch size, and K the representation dimension. In MV-HJEPA we enforce $K = 2d$ and interpret each representation as a phase-space state $z = [q; p] \in \mathbb{R}^{2d}$ with $q, p \in \mathbb{R}^d$ (channel-wise split, `split_qp=true`). We write $\text{concat}_v x^{(v)}$ for concatenation of views along the batch dimension, and reshape for a view/batch reshape with the same underlying memory order as the implementation.

Algorithm 1 Multi-crop view generation (used by `CIFAR100MultiCrop` and `ImageNetMultiCrop`)

Input: Image I, multi-crop config: (V_g, V_ℓ), output size s, scales (α_g, β_g) and $(\alpha_\ell, \beta_\ell)$
Output: List of views $\{x^{(v)}\}_{v=1}^{V}$ where $V = V_g + V_\ell$

Initialize empty list `views` $\leftarrow [\,]$
for $i = 1$ to V_g **do**
  $x \leftarrow \text{Augment}(I; \text{RandomResizedCrop}(s, \text{scale} = [\alpha_g, \beta_g]), \text{HFlip}, \text{CJ}, \text{Gray}, \text{Blur}, \text{Solarize})$
  Append x to `views`
end for
for $i = 1$ to V_ℓ **do**

7: $x \leftarrow \text{Augment}(I;\ \text{RandomResizedCrop}(s, \text{scale} = [\alpha_\ell, \beta_\ell]), \text{HFlip}, \text{CJ}, \text{Gray}, \text{Blur}, \text{Solarize})$
8: Append x to `views`
9: **end for**
10: **return** `views`

Implementation note (dataset-specific): CIFAR views are produced as float tensors with normalization applied in the dataset pipeline; ImageNet views are emitted as compact `uint8` tensors (via `PILToTensor`) and normalized on-device in the training loop.

Algorithm 2 ResNet token encoder forward pass (token mode, optional q/p split)

Input: Image batch $x \in \mathbb{R}^{B\times 3\times H\times W}$, ResNet stem+layers, token layer $\ell \in \{\texttt{layer2}, \texttt{layer3}, \texttt{layer4}\}$, channel projection to d_f, optional pooling to `token_hw` $= h = w$, `split_qp` flag
Output: Representation $z \in \mathbb{R}^{B\times K}$ where $K = h \cdot w \cdot d_f$
$u \leftarrow \text{stem}(x)$
$u \leftarrow \text{layer1}(u)$
$u \leftarrow \text{layer2}(u)$; set fmap $\leftarrow u$ if $\ell =$ `layer2`
$u \leftarrow \text{layer3}(u)$; set fmap $\leftarrow u$ if $\ell =$ `layer3`
$u \leftarrow \text{layer4}(u)$; set fmap $\leftarrow u$ if $\ell =$ `layer4`
if `token_hw` is set **then**
  fmap $\leftarrow \text{AdaptiveAvgPool2d}(\text{fmap}, (h, w))$
end if
$t \leftarrow \text{Conv1}\times 1 + \text{BN}(\text{fmap}) \in \mathbb{R}^{B\times d_f\times h\times w}$
$t \leftarrow \text{permute}(t) \in \mathbb{R}^{B\times h\times w\times d_f}$
if `split_qp` = false **then**
  $z \leftarrow \text{flatten}(t) \in \mathbb{R}^{B\times(hwd_f)}$
else
  Split channels: $t \mapsto (t_q, t_p)$ with $t_q, t_p \in \mathbb{R}^{B\times h\times w\times(d_f/2)}$
  $q \leftarrow \text{flatten}(t_q) \in \mathbb{R}^{B\times(hw(d_f/2))}$; $p \leftarrow \text{flatten}(t_p) \in \mathbb{R}^{B\times(hw(d_f/2))}$
  $z \leftarrow [q; p] \in \mathbb{R}^{B\times(hwd_f)}$
end if
return z

Algorithm 3 MV-HJEPA training step (matches `train_*_hamjepa.py` when `hjepa:` is present)

Input: Views $\{x^{(v)}\}_{v=1}^{V}$, encoder E, projector P (identity for MV-HJEPA), predictor Flow_H, loss/regularizer weights $\lambda_\bullet$.
Input: MV-HJEPA requires `split_qp=true` and $K = 2d$
$x_{\text{cat}} \leftarrow \text{concat}_v\, x^{(v)} \in \mathbb{R}^{(VB)\times C\times H\times W}$
$z_{\text{cat}} \leftarrow P(E(x_{\text{cat}})) \in \mathbb{R}^{(VB)\times 2d}$
$z_{\text{views}} \leftarrow \text{reshape}(z_{\text{cat}}) \in \mathbb{R}^{V\times B\times 2d}$
$z_0 \leftarrow z_{\text{views}}[0]$, $z_1 \leftarrow z_{\text{views}}[1]$
if CIFAR-100 **then**
  $\mathcal{L}_{\text{pred}} \leftarrow \text{HamConsistency}(z_0, z_1)$ with the configured internal `bidirectional` flag
else ▷ ImageNet-100: internal `bidirectional=false`
  $\mathcal{L}_{\text{pred}} \leftarrow \text{HamConsistency}(z_0, z_1)$
  if `bidirectional` **then**
    $\mathcal{L}_{\text{pred}} \leftarrow \frac{1}{2}\left(\mathcal{L}_{\text{pred}} + \text{HamConsistency}(z_1, z_0)\right)$
  end if
end if
$z_{\text{all}} \leftarrow z_{\text{views}}.\text{reshape}(VB, 2d)$
Split $z_{\text{all}} \mapsto (q_{\text{all}}, p_{\text{all}})$ along the last dimension
$\mathcal{L}_{\text{budget}} \leftarrow \text{EnergyBudget}(z_{\text{all}})$
$\mathcal{L}_{\text{var}} \leftarrow \text{VarFloor}(q_{\text{all}})$ (+ $\text{VarFloor}(p_{\text{all}})$ on CIFAR-100)
$\mathcal{L}_{\log\det} \leftarrow \text{LogDetFloor}(q_{\text{all}}) + \text{LogDetFloor}(p_{\text{all}})$
$\mathcal{L}_{\text{mean}} \leftarrow \text{mean}_{\text{dim}}\left(\text{mean}_{\text{batch}}(z_{\text{all}})^{\odot 2}\right)$ ▷ `mean(dim=0).square().mean()`
$\mathcal{L} \leftarrow \mathcal{L}_{\text{pred}} + \lambda_{\text{budget}}\mathcal{L}_{\text{budget}} + \lambda_{\text{var}}\mathcal{L}_{\text{var}} + \lambda_{\log\det}\mathcal{L}_{\log\det} + \lambda_{\text{mean}}\mathcal{L}_{\text{mean}}$

20: Update parameters of $\{E, P, \mathrm{Flow}_H\}$ with AdamW; optionally clip gradients and step LR schedule

Algorithm 4 Hamiltonian Consistency loss (`HamiltonianConsistencyLoss`)

Input: States $z_a, z_b \in \mathbb{R}^{B\times 2d}$, predictor Flow_H, options: `detach_target`, `match` $\in \{\mathtt{q}, \mathtt{qp}\}$, `p_weight`, `energy_weight`, `bidirectional`
Output: Scalar loss $\mathcal{L}$

$\widehat{z}_{a\to b} \leftarrow \mathrm{Flow}_H(z_a; \mathrm{direction} = +1)$
$z_b^\star \leftarrow z_b$ if not `detach_target` else $\mathrm{stopgrad}(z_b)$
if `match=qp` **then**
  $\ell_{a\to b} \leftarrow \mathrm{MSE}(\widehat{z}_{a\to b}, z_b^\star)$
else ▷ `match=q`
  $\ell_{a\to b} \leftarrow \mathrm{MSE}(\widehat{z}_{a\to b}[:, :d], z_b^\star[:, :d])$
  if `p_weight` > 0 **then**
    $\ell_{a\to b} \leftarrow \ell_{a\to b} + \mathtt{p_weight} \cdot \mathrm{MSE}(\widehat{z}_{a\to b}[:, d:], z_b^\star[:, d:])$
  end if
end if
if `energy_weight` > 0 **then**
  Split $z_a \mapsto (q_a, p_a)$ and $\widehat{z}_{a\to b} \mapsto (\hat{q}, \hat{p})$
  $H_a \leftarrow \mathrm{stopgrad}\big(H(q_a, p_a)\big)$; $H_{\hat{z}} \leftarrow H(\hat{q}, \hat{p})$
  $\ell_{a\to b} \leftarrow \ell_{a\to b} + \mathtt{energy_weight} \cdot \mathrm{MSE}(H_{\hat{z}}, H_a)$
end if
$\mathcal{L} \leftarrow \ell_{a\to b}$
if `bidirectional` **then**
  Compute $\ell_{b\to a}$ identically using $\mathrm{direction} = -1$ and target z_a (optionally detached)
  $\mathcal{L} \leftarrow \frac{1}{2}(\ell_{a\to b} + \ell_{b\to a})$
end if
return $\mathcal{L}$

Algorithm 5 Hamiltonian Flow Predictor (forward pass, `HamiltonianFlowPredictor`)

Input: State $z_0 \in \mathbb{R}^{\cdots\times 2d}$, Hamiltonian module H, method, steps S, base step size $\Delta t > 0$, direction $s \in \{+1, -1\}$, `integrate_fp32`, `learn_dt`
Output: State $z_T \in \mathbb{R}^{\cdots\times 2d}$

Split $z_0 \mapsto (q_0, p_0)$
$\Delta t \leftarrow s \cdot \Delta t$; if `learn_dt`, set $\Delta t \leftarrow s \cdot \mathrm{softplus}(\mathtt{raw_dt})$
if `integrate_fp32` enabled **and** $\mathrm{dtype}(z_0) \in \{\mathrm{fp16}, \mathrm{bf16}\}$ **then**
  Cast (q_0, p_0) to fp32
end if
if H is separable **and** method is leapfrog **then**
  $(q_T, p_T) \leftarrow \mathrm{IntegrateSeparableLeapfrog}(V_\phi, q_0, p_0, \Delta t, S)$
else
  $(q_T, p_T) \leftarrow \mathrm{IntegrateHamiltonian}(H, q_0, p_0, \Delta t, S, \mathrm{method})$
end if
$z_T \leftarrow [q_T; p_T]$
if `integrate_fp32` enabled **and** original dtype was bf16/fp16 **then**
  Cast z_T back to the original dtype
end if
return z_T

Algorithm 6 Symplectic integration routines (as in `hamjepa/integrators.py`)

Input: General Hamiltonian $H(q, p)$ or separable potential $V(q)$, initial (q, p), step size Δt, steps S, method $\in \{\mathtt{leapfrog}, \mathtt{symplectic_euler}\}$
Output: Final (q, p) after S steps

function INTEGRATEHAMILTONIAN($H, q, p, \Delta t, S,$ method)
  for $n = 1$ to S **do**
    if method = symplectic_euler **then**
      $p \leftarrow p - \Delta t\, \nabla_q H(q, p)$

$q \leftarrow q + \Delta t\, \nabla_p H(q,p)$ ▷ ∇_p uses updated p
else ▷ method = leapfrog (velocity Verlet)
$p_{1/2} \leftarrow p - \frac{\Delta t}{2} \nabla_q H(q,p)$
$q \leftarrow q + \Delta t\, \nabla_p H(q,p_{1/2})$
$p \leftarrow p_{1/2} - \frac{\Delta t}{2} \nabla_q H(q,p_{1/2})$
end if
end for
return (q,p)
end function
function INTEGRATESEPARABLELEAPFROG($V, q, p, \Delta t, S$)
for $n = 1$ to S **do**
$p_{1/2} \leftarrow p - \frac{\Delta t}{2} \nabla_q V(q)$
$q \leftarrow q + \Delta t\, p_{1/2}$
$p \leftarrow p_{1/2} - \frac{\Delta t}{2} \nabla_q V(q)$ ▷ $\nabla_q V$ uses updated q
end for
return (q,p)
end function

Algorithm 7 Phase-space energy budget regularizer (`PhaseSpaceEnergyBudget`)

Input: States $z \in \mathbb{R}^{\cdots \times 2d}$, targets $(q_{\text{tgt}}, p_{\text{tgt}})$, `ddp_sync`
Output: Scalar loss $\mathcal{L}_{\text{budget}}$

Flatten $z \to z_f \in \mathbb{R}^{N \times 2d}$ and cast to fp32
Split $z_f \mapsto (q,p)$ with $q, p \in \mathbb{R}^{N \times d}$
$q2_sum \leftarrow \sum q^{\odot 2}$, $p2_sum \leftarrow \sum p^{\odot 2}$, `count` $\leftarrow Nd$
if `ddp_sync` and DDP initialized **then**
All-reduce (`SUM`) $q2_sum$, $p2_sum$, and `count`
end if
$q2_\mu \leftarrow q2_sum/$`count`; $p2_\mu \leftarrow p2_sum/$`count`
$\mathcal{L}_{\text{budget}} \leftarrow (q2_\mu - q_{\text{tgt}})^2 + (p2_\mu - p_{\text{tgt}})^2$
return $\mathcal{L}_{\text{budget}}$

Algorithm 8 Variance floor regularizer (`VarianceFloor`)

Input: Features $x \in \mathbb{R}^{N \times d}$, std floor $\sigma_{\min}$, ε, `ddp_sync`
Output: Scalar loss $\mathcal{L}_{\text{var}}$

Cast x to fp32; if $N < 2$ return 0
Compute sufficient stats: $s_1 \leftarrow \sum_n x_n$, $s_2 \leftarrow \sum_n x_n^{\odot 2}$, and $n \leftarrow N$
if `ddp_sync` and DDP initialized **then**
All-reduce (`SUM`) s_1, s_2, and n
end if
$\mu \leftarrow s_1/n$; var $\leftarrow s_2/n - \mu^{\odot 2}$; $\sigma \leftarrow \sqrt{\max(\text{var}, 0) + \varepsilon}$
$\mathcal{L}_{\text{var}} \leftarrow \frac{1}{d} \sum_{i=1}^{d} \text{ReLU}(\sigma_{\min} - \sigma_i)^2$
return $\mathcal{L}_{\text{var}}$

Algorithm 9 Projected log-det floor (`ProjectedLogDetFloor`)

Input: Features $x \in \mathbb{R}^{B \times D}$, projection dim k, log-det floor $\ell_{\min}$, ε, refresh interval r, optional `pr_norm_floor` and `eigmax_frac_ceiling`, `ddp_sync`
Output: Scalar loss $\mathcal{L}_{\log\det}$

Cast x to fp32; if $B < 2$ return 0
$k \leftarrow \min(k, D, B-1)$
if need resample (first call / device change / shape change / step $\bmod r = 0$) **then**
Sample $R \sim \mathcal{N}(0,1)^{D \times k}$ and orthonormalize columns via QR: $R \leftarrow \text{qr}(R)$
end if
Center and project: $x_c \leftarrow x - \text{mean}(x)$; $y \leftarrow x_c R \in \mathbb{R}^{B \times k}$
Compute sufficient stats: `sum_y` $\leftarrow \sum_b y_b$, `sum_yy` $\leftarrow y^\top y$, $n \leftarrow B$
if `ddp_sync` and DDP initialized **then**

All-reduce (SUM) $\texttt{sum_y}$, $\texttt{sum_yy}$, and n
end if
$n \leftarrow \max(n, 2)$; $\mu_y \leftarrow \texttt{sum_y}/n$
$\Sigma \leftarrow \frac{\texttt{sum_yy} - n \cdot (\mu_y \mu_y^\top)}{n-1}$; $\Sigma \leftarrow \frac{1}{2}(\Sigma + \Sigma^\top) + \varepsilon I$
$\ell \leftarrow \frac{1}{k} \log\det(\Sigma)$ ▷ implemented via `slogdet`
$\mathcal{L}_{\log\det} \leftarrow \mathrm{ReLU}(\ell_{\min} - \ell)^2$
Compute $\mathrm{PR} = \frac{\mathrm{tr}(\Sigma)^2}{\mathrm{tr}(\Sigma^2)}$, $\mathrm{pr_norm} = \mathrm{PR}/k$, and $\mathrm{eigmax_frac} = \lambda_{\max}/\mathrm{tr}(\Sigma)$
if `pr_norm_floor` is set **then**
    $\mathcal{L}_{\log\det} \leftarrow \mathcal{L}_{\log\det} + \mathrm{ReLU}(\texttt{pr_norm_floor} - \mathrm{pr_norm})^2$
end if
if `eigmax_frac_ceiling` is set **then**
    $\mathcal{L}_{\log\det} \leftarrow \mathcal{L}_{\log\det} + \mathrm{ReLU}(\mathrm{eigmax_frac} - \texttt{eigmax_frac_ceiling})^2$
end if
return $\mathcal{L}_{\log\det}$

Algorithm 10 SIGReg forward pass (random slicing + Epps–Pulley statistic)

Input: Batch $z \in \mathbb{R}^{\cdots \times N \times D}$, slices K, knots $t_1, \dots, t_T$ (with weights w), refresh interval r, `ddp_sync`
Output: Scalar statistic $S(z)$ (mean-reduced in our configs)

(Slicing) If step $\bmod r = 0$, sample $A \in \mathbb{R}^{D \times K}$ with i.i.d. $\mathcal{N}(0,1)$ columns and normalize each column to unit norm
if `ddp_sync` and DDP initialized **then**
    Ensure all ranks share the same sampling seed for A (broadcast seed from rank 0)
end if
Project: $y \leftarrow zA \in \mathbb{R}^{\cdots \times N \times K}$
(Epps–Pulley) Compute empirical CF on each slice:
    $\hat{c}(t) = \frac{1}{N}\sum_n \cos(t\, y_n)$, $\hat{s}(t) = \frac{1}{N}\sum_n \sin(t\, y_n)$ (optionally globally averaged via all-reduce if `ddp_sync`)
    Target CF for $\mathcal{N}(0,1)$ is $\phi(t) = e^{-t^2/2}$
    $S_k \leftarrow N \sum_{j=1}^{T} w_j \Big[(\hat{c}(t_j) - \phi(t_j))^2 + \hat{s}(t_j)^2 \Big]$ for each slice k
Reduce over slices (and leading dims) by mean: $S(z) \leftarrow \mathrm{mean}_k(S_k)$
return $S(z)$

Algorithm 11 LeJEPA baseline step with SIGReg regularization (non-MV branch in `train_*_hamjepa.py`)

Input: Views $\{x^{(v)}\}_{v=1}^{V}$, encoder E, projector P, number of global views V_g, regularizer $\mathcal{R}$ (SIGReg), weight λ_{reg}

$x_{\text{cat}} \leftarrow \mathrm{concat}_v\, x^{(v)}$
$z_{\text{cat}} \leftarrow P(E(x_{\text{cat}})) \in \mathbb{R}^{(VB)\times D}$
$z_{\text{views}} \leftarrow \mathrm{reshape}(z_{\text{cat}}) \in \mathbb{R}^{V \times B \times D}$
$\bar{z} \leftarrow \frac{1}{V_g}\sum_{v=1}^{V_g} z_{\text{views}}[v] \in \mathbb{R}^{B\times D}$ ▷ mean of global views
$\mathcal{L}_{\text{pred}} \leftarrow \frac{1}{2}\,\mathrm{mean}\big(\|z_{\text{views}} - \bar{z}\|_2^2\big)$ ▷ matches `lejepa_prediction_loss`
$\mathcal{L}_{\text{reg}} \leftarrow \mathcal{R}(z_{\text{views}})$ ▷ SIGReg expects shape $(\cdots, N, D)$
$\mathcal{L} \leftarrow \mathcal{L}_{\text{pred}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}$ ▷ spectral term omitted here (zero in our SIGReg configs)
Update $\{E, P\}$ with AdamW; optionally step LR schedule and log diagnostics

H.12 YAML Configuration Knobs

The repository is fully config-driven: the training scripts read a YAML file and route into either (i) MV-HJEPA mode when the `hjepa:` stanza is present (and `split_qp=true`), or (ii) the LeJEPA+regularizer baseline when `hjepa:` is absent (e.g. `regularizer.type=sigreg`). Below we document every key appearing in the provided configs, exactly as it is consumed by the code. The reference configs use `seed=42`; the replicated 80-epoch CIFAR-100 suite additionally uses `seed=43`.

H.12.1 Data (`data:`)

- `root`: Dataset root path passed to `CIFAR100` / `ImageFolder` loader.
- `batch_size`: DataLoader batch size (`batch_size=batch_size`). We use larger batches on CIFAR (512) and smaller on ImageNet (256) to balance GPU memory and throughput.
- `num_workers`: DataLoader worker count. Higher values improve input pipeline throughput when CPU-bound.
- `drop_last`: If true, drops the last incomplete batch. This avoids smaller last batches, which can destabilize batch statistics and covariance-based regularizers (and helps DDP batch-size consistency).
- `pin_memory` (ImageNet configs): Enables pinned host memory to speed CPU→GPU transfers.
- `persistent_workers` (ImageNet configs): Keeps workers alive across epochs to reduce worker spin-up overhead.

- `prefetch_factor` (ImageNet configs): Worker prefetch depth; increases overlap between CPU augmentation and GPU training.
- `num_global_views`, `num_local_views`: Multi-crop configuration. In all provided runs we set two global views and no local views (`2/0`), matching the two-view training objective in both MV-HJEPA and the LeJEPA baseline.
- `out_size` (ImageNet configs): Output crop resolution for `RandomResizedCrop`. We use 224 (standard ImageNet scale).
- `global_scale`, `local_scale` (ImageNet configs): Scale ranges for global/local crops. Only global crops are used when `num_local_views=0`.
- *CIFAR note:* the CIFAR YAMLs omit `out_size/global_scale/local_scale`, so the dataset defaults are used (`out_size=32`, `global_scale=(0.3,1.0)`, `local_scale=(0.05,0.3)`).

H.12.2 Model (`model:`)

- `encoder_mode`: `global` uses GAP (global average pooling)+FC; `tokens` exposes an intermediate feature map and flattens it into tokens. All provided configs use `tokens`.
- `encoder_stem`: `cifar` modifies ResNet-18 stem (3×3 stride 1, no maxpool) for 32×32 inputs; `imagenet` keeps the default 7×7 stride 2 + maxpool stem.
- `token_layer`: Which ResNet stage to expose as the token grid (`layer2/layer3/layer4`). We use `layer3`, giving an 8×8 grid on CIFAR and a 14×14 grid on ImageNet before any optional pooling.
- `token_d_f`: Number of channels after the 1×1 token projection (`Conv1x1+BN`). In token mode, the representation dimension is $K = h \cdot w \cdot$ `token_d_f`.
- `token_hw` (ImageNet configs): Optional adaptive pooling size for the token grid. We set `token_hw=8` on ImageNet so that with `token_d_f=32` we obtain $K = 8 \cdot 8 \cdot 32 = 2048$ (matching `embed_dim`).
- `embed_dim`: Output dimension K of the encoder in token mode. For our settings this is tied to the token grid geometry (e.g. $8 \cdot 8 \cdot 32 = 2048$).
- `proj_hidden`, `proj_out`: Hidden/output sizes for the optional MLP projector. These are read from the YAML but are only used when `projector_type=mlp`.
- `projector_type`: `identity` returns z unchanged; `mlp` applies `MLPProjector`. MV-HJEPA requires `identity` so the channel-wise q/p semantics are preserved.
- `split_qp`: If true, the token channels are split into (q, p) halves per token and concatenated, enforcing $K = 2d$ and enabling the phase-space interpretation required by MV-HJEPA.

H.12.3 MV-HJEPA predictor (`hjepa:`)

(Only present in `*_hjepa_mv.yaml` configs.)

- `hamiltonian`: Choice of Hamiltonian parameterization. We use `separable`, i.e. $H(q,p) = \frac{1}{2}\|p\|^2 + V_\phi(q)$, to enable explicit symplectic leapfrog integration.
- `method`: Integrator choice. We use `leapfrog` (velocity Verlet), implemented in `integrate_separable_leapfrog` for separable H.
- `steps`: Number of integrator steps S per forward pass. We use 2 to keep compute low while still giving a nontrivial flow.
- `dt`: Fixed step size Δt. Used as a constant buffer when `learn_dt=false`.
- `learn_dt`: If true, Δt becomes learnable via a softplus parameterization. MV-HJEPA disables this (`false`) and the training script explicitly rejects learnable-dt to avoid degenerate solutions ("dt-collapse").
- `hidden_dim`, `depth`: MLP size/depth used inside the Hamiltonian residual network (for V_ϕ in the separable case).
- `residual_scale`: Multiplier on the Hamiltonian residual output. This is a stability knob: the code is designed so the Hamiltonian has a quadratic base plus a small learned residual.

- `residual_scale_warmup_epochs`: In the CIFAR-100 loop, scales `residual_scale` by $\min(1, e/10)$ at epoch $e \geq 1$ (implemented by overwriting `predictor.H.residual_scale` each epoch). The ImageNet-100 loop does not apply this field and uses the fixed configured residual scale.
- `base_coeff`: Coefficient on the quadratic base potential term in $V(q)$ (i.e. $0.5\,$`base_coeff`$\,\|q\|^2$). A nonzero coefficient supplies a quadratic base, but the learned residual can modify or partially cancel its force.
- `integrate_fp32`: If true, the predictor casts z to fp32 during integration when training in bf16/fp16, improving numerical stability of gradient-based integration.

H.12.4 MV-HJEPA loss (`loss:`)

- `match`: Which part of the state to match in the consistency loss: `q` matches positions with an auxiliary p penalty when `p_weight`>0 (the CIFAR-100 value is 0.2), while `qp` matches the full state.
- `p_weight`: Only used when `match=q`. Adds an auxiliary MSE on p with weight `p_weight`.
- `detach_target`: If true, the target branch is detached in the prediction loss (safer default to prevent the target moving to satisfy the predictor).
- `energy_weight`: Optional extra term matching $H(z)$ before/after the flow. Set to `0.0` in all provided configs (disabled).
- `bidirectional`: If true, averages both view orders ($z_0 \to z_1$ and $z_1 \to z_0$). In the ImageNet training script this averaging is done explicitly in the loop with positive step size in both orders; in the CIFAR script it is handled inside the loss module with negative step size for the return order.

H.12.5 MV-HJEPA anti-collapse regularizers (`regularizer:`)

(These are the *phase-space* regularizers used in the MV-HJEPA branch; they are not the SIGReg baseline.)

- `q_per_dim_target`, `p_per_dim_target`: Targets for the phase-space energy budget: they control the desired per-dimension mean-square magnitude of q and p.
- `q_std_floor`: Per-dimension standard deviation floor used by `VarianceFloor` on q (and on p in the CIFAR MV script; ImageNet MV applies the variance floor to q only).
- `ddp_sync`: If true and DDP is initialized, the sufficient statistics for the budget/variance/logdet terms are aggregated across ranks using all-reduce. This makes the regularizers behave as if computed on the global batch.
- `q_logdet_proj_dim`, `p_logdet_proj_dim`: Projection dimension k for the log-det regularizer on q and p.
- `q_logdet_floor`, `p_logdet_floor`: Minimum allowed log-determinant per projected dimension (a volume floor) used by `ProjectedLogDetFloor`.
- `q_logdet_eps`, `p_logdet_eps`: Diagonal jitter εI added to the covariance before computing `slogdet`.
- `q_logdet_refresh_interval`, `p_logdet_refresh_interval`: How often the random projection matrix is resampled (larger values reduce overhead and noise by reusing projections).
- `q_pr_norm_floor`, `p_pr_norm_floor`: Optional lower bounds on normalized participation ratio PR/k (encourages higher effective rank).
- `q_eigmax_frac_ceiling`, `p_eigmax_frac_ceiling`: Optional upper bounds on $\lambda_{\max}/\mathrm{tr}(\Sigma)$ to prevent a single dominant direction.

H.12.6 Training (`train:`)

- `epochs`: Total training epochs.
- `lr`: Base AdamW learning rate for encoder/projector (and predictor in CIFAR MV).

- `h_lr` (ImageNet MV config): Separate learning rate used specifically for the Hamiltonian parameters inside the predictor (`predictor.H.parameters()`); the ImageNet training script builds a two-group optimizer when these parameters exist.
- `weight_decay`: AdamW weight decay for the main optimizer.
- `warmup_epochs`: Warmup duration used by the linear warmup + cosine decay scheduler (when enabled by the script).
- `min_lr_ratio`: Sets $\eta_{\min} = \texttt{lr} \cdot \texttt{min_lr_ratio}$ for cosine annealing.
- `precision`: Mixed precision mode. `bf16` enables CUDA autocast bf16 in training.
- `ckpt_dir`: Output directory for checkpoints.
- `checkpoint_every` (ImageNet configs): Save checkpoints every k epochs (ImageNet script only).
- `grad_clip`: If > 0, applies global-norm gradient clipping (`clip_grad_norm_`). The CIFAR-100 parameter list includes the predictor; the ImageNet-100 clipping call excludes the separately grouped Hamiltonian parameters.
- `log_every`: Print training diagnostics every `log_every` steps.
- `profile_timing`, `profile_every` (ImageNet MV config): Optional lightweight profiling of data time vs step time (only used in the ImageNet script).
- **MV-HJEPA loss weights:** `lambda_budget`, `lambda_var`, `lambda_logdet`, `lambda_mean` weight the corresponding regularizers in Algorithm 3.
- **SIGReg baseline weights:** `lambda_reg` is the coefficient on the SIGReg term in Algorithm 11.
- `h_update_interval`, `h_grad_clip` (SIGReg YAMLs): These knobs are parsed by the training scripts for the *learnable Hamiltonian regularizer* path (`regularizer.type=ham_sigreg` with learnable H). In the provided SIGReg-only configs (`type=sigreg`), there are no H parameters, so these settings are effectively inert.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the marginal-geometry theory, the canonical Hamiltonian lift under explicit oracle assumptions, and the controlled HamJEPA experiments. The matched controls rule out attributing the empirical gain uniquely to symplecticity or reversibility. The theory claims are supported by Section 3 and the appendix; the empirical claims are supported by Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a limitations paragraph in Section 6, noting that this is a controlled geometry study rather than a SOTA SSL recipe, that the main baseline is SIGReg, that ImageNet-100 is single-seed, and that the downstream operator H is not learned. It also states that symplecticity and reversibility are not individually necessary in the tested controls, and that learned-marginal baselines and full-recipe generalization remain untested.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The theoretical statements specify their assumptions in the theorem/lemma statements. Complete proofs are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental protocol, datasets, view construction, backbone, projector choice, optimization setup, and evaluation metrics are described in Section 5 and Appendix H. Hyperparameters and additional implementation details are provided in the appendix. Appendix F.4 specifies the exact additional-control splitting coefficients, preprocessing, probe settings, and uncertainty definitions. The accompanying archive provides source snapshots, configurations, and recorded commands for reproducing the additional experiments. Appendix H distinguishes these reproducibility details from the unresolved exact producing revision of the historical ImageNet-100 checkpoint.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The public implementation is linked in the introduction. The accompanying supplementary archive provides the source snapshots, configurations, recorded commands, evaluation definitions, and saved outputs for the additional CIFAR-100 experiments. The code specifies dataset access and preparation; dataset images are not redistributed and ImageNet access remains subject to its terms. Appendix H states the archive scope and the unresolved historical source-to-checkpoint provenance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 describes the controlled protocol, datasets, readouts, and evaluation metrics. Training and hyperparameter details, including the active loss weights and dataset-specific matching and warmup semantics, are provided in Appendix H; the exact additional-control splitting coefficients and evaluation definitions are provided in Appendix F.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: CIFAR-100 30-epoch results report mean and standard deviation over three seeds. The additional 80-epoch CIFAR-100 comparisons use two independent pretraining seeds; reported error bars are descriptive sample standard deviations across those runs, with each linear-probe value first averaged over five probe optimizations. The original 80-epoch results, ImageNet-100 results, and single-seed sensitivity conditions are explicitly distinguished. No formal significance is claimed from two pretraining seeds, and correlated metric/k sweeps are not independent replications.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix H reports the hardware, training time, batch sizes, number of epochs, and evaluation resources used for the main experiments. Historical ImageNet-100 timings are reported separately for HamJEPA and SIGReg as notebook-cell wall times. The additional matched A100 preflight benchmark reports cold per-step time, throughput, parameter overhead, and peak allocated memory from one timed step per method and seed, with no warmup; these measurements do not establish steady-state throughput.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is a foundational representation-learning study on standard image benchmarks and does not involve human-subject experiments, scraped private data, security vulnerabilities, or high-risk deployment claims.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: HamJEPA may improve controlled visual representation learning. Potential negative impacts are generic to improved visual representations, including downstream misuse in surveillance, biometric identification, or biased high-stakes classifiers; deployment should require dataset documentation, privacy review, and fairness evaluation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk generative model, surveillance system, scraped dataset, security exploit, or dual-use capability requiring special access controls.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper credits the original datasets and PyTorch, and Appendix H gives the verified software license links and ImageNet access terms. The official CIFAR distribution page does not state a separate named dataset license, and the supplied artifacts do not pin all runtime versions. No datasets are redistributed.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The paper does not introduce a new dataset or release model weights as a required research artifact. The accompanying source and result archive documents the additional controls, evaluation definitions, configurations, recorded commands, and provenance limitations. The public implementation's license and LeJEPA attribution are identified, but the recovered source snapshots do not include a complete statement of licensing scope for every archived component.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve human-subject experiments, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not an important, original, or non-standard component of the core method or experiments. Any use for writing, editing, or formatting does not affect the scientific method or results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.