GEOMETRY IS NOT ENOUGH: SYMPLECTIC STRUCTURE AND NOISE ROBUSTNESS IN JEPA WORLD MODELS FOR BEAM DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Joint-Embedding Predictive Architectures (JEPAs) learn dynamics in latent space and promise cheaper, noise-robust learning-based surrogates for physical simulation. Without a reconstruction loss anchoring the representation, however, the encoder and predictor are free to settle on latents that are mutually consistent with little knowledge about the system dynamics. But whether enforcing that structure in latent space improves performance remains poorly understood. We address this by studying JEPAs on a system whose latent dynamics have a known target: longitudinal beam dynamics, whose one-turn map is a symplectic Hamiltonian flow, and use that structure as a reference for the learned latent. Our contributions are threefold: we examine whether enforcing symplectic structure on the latent predictor restores the geometry of the true one-turn map and improves accuracy, benchmark JEPA against reconstruction-based and pixel-space surrogates in error and cost, and assess its robustness to noise and distribution shift. On $23{,}101$ BLonD simulations of the FNAL Recycler, we find that symplectic predictors restore the geometry of the one-turn map but only direct map parameterisations improve accuracy, and that JEPA trains $2.1\times$–$2.5\times$ faster, and filters input noise, though reconstruction retains the lowest error. Our findings align with prior work on transfer-operator and latent Hamiltonian learning and point to encoder objectives and training regimes that would make JEPA a reliable surrogate for complex physical dynamics.

1 INTRODUCTION

Learning the dynamics of complex physical systems is central to forecasting, fault detection and diagnosis, and feedback control. Consider the longitudinal dynamics of a particle bunch in a synchrotron. During each turn, the bunch passes through radio-frequency cavities that confine it in a potential bucket, and every particle executes a slow synchrotron oscillation in the phase–energy plane whose frequency is set by the machine settings. The bunch is therefore a density in a two-dimensional phase space, advanced once per turn by a map that is deterministic, autonomous, and symplectic. Operational quantities such as bunch centroid, length, emittance, and intensity are functionals of this density and must be tracked over thousands of turns to detect instabilities, diagnose mismatch, and drive low-level RF feedback. This structure is not specific to accelerators: weather, fluid, and power-grid models likewise advance a high-dimensional state under a fixed physical rule and read out a few operational quantities from it (Figure 1). Such tasks require models that capture the dynamics accurately while remaining reliable under changing initial conditions, external perturbations, and imperfect observations.

High-fidelity particle beam dynamics simulators (e.g., BLonD (Timko et al., 2022)) are not differentiable and must be queried repeatedly in optimization and control loops, while differentiable frameworks (e.g., Cheetah (Kaiser et al., 2024b)) enable gradient-based use but remain bound to the discretization, so resolving fine phase-space scales over long horizons is still prohibitively expensive for real-time inference. Deep learning has emerged as a viable alternative. Learned alternatives fall into two families (Figure 1b). *Observation-space predictors* map the current observation directly to the next one: data-driven surrogates (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021; Brandstetter et al., 2022), neural operators (Li et al., 2021b; Lu et al., 2021), and PINNs (Raissi et al.,

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

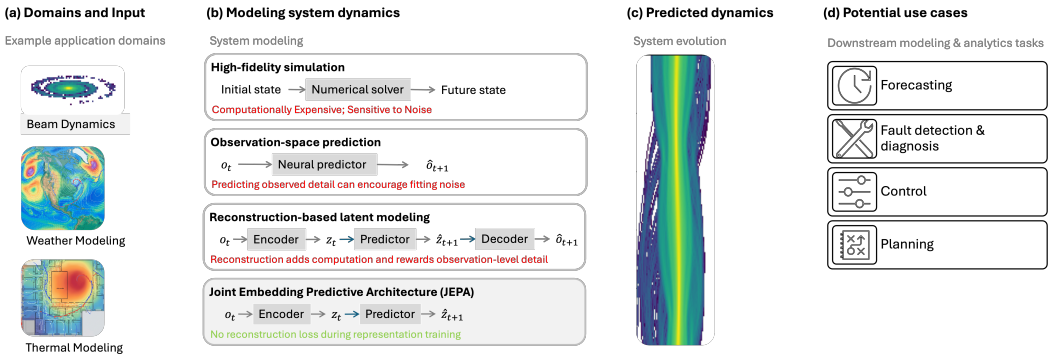

Figure 1: **Overview.** Predicting physical dynamics (a,c) for downstream tasks (d) trades cost for fidelity (b); JEPA avoids reconstruction, but its efficacy for dynamics modeling is largely unknown.

2019). Because they operate on the full spatially resolved state, their cost scales with resolution, autoregressive rollout accumulates error (Lippe et al., 2023), and matching every observed pixel encourages fitting noise rather than dynamics. *Reconstruction-based latent models* instead encode the observation, evolve a compressed state, and decode it back (Wiewel et al., 2019; Vlachas et al., 2022). This removes the resolution dependence but the decoder adds computation at every step, and the reconstruction loss again rewards observation-level detail that need not be predictively relevant.

In contrast, Joint-Embedding Predictive Architecture (JEPA) predicts a target representation from an encoded context rather than reconstructing the target in observation space (LeCun et al., 2022). This means the predictive objective is to match learned features as opposed to observations. Originally developed for self-supervised visual learning (Assran et al., 2023; Bardes et al., 2024), JEPA-style latent world models have since been used for planning and control (Sobal et al., 2022; Garrido et al., 2024; Zhou et al., 2024; Assran et al., 2025) and as latent state-space models of physical systems (Ulmen et al., 2025). However, there has not been any formal study into the suitability of JEPA models for scientific simulation, wherein predictions must agree quantitatively with governing dynamics and conservation laws, retain fast, low-amplitude modes, and tolerate sparse or noisy observations. Whether the latent dynamics a JEPA learns reflect the structure of the underlying system remains poorly understood.

In the present work, we study JEPA for modeling complex physical dynamics, with longitudinal beam simulations of the FNAL Recycler as the target application. We validate JEPA along three axes: (1) whether symplectic predictors restore the geometry of the one-turn map and improve accuracy, (2) how it compares with reconstruction-based and neural-operator baselines on error and cost, and (3) robustness to distribution shift and observation noise. We do so by ablating an MLP predictor against five structure-preserving architectures within a fixed JEPA state-space model (Ulmen et al., 2025), then benchmarking the best variants against non-JEPA counterparts. To support this, we generate a dataset of BLonD simulations of the FNAL Recycler spanning Gaussian and held-out non-Gaussian bunch profiles, each tracked over hundreds of turns and rendered as longitudinal phase-space density frames. Our study reveals key architectural considerations for JEPA as a physical world model of complex system dynamics and exposes structural requirements on the encoder space that the standard objective leaves implicit.

2 Related Work

Numerical simulation advances spatiotemporal dynamics at a cost fixed by the discretization (for particle-in-cell beam codes, by macroparticle count and grid; Qiang et al., 1999), which makes many-query workflows prohibitively expensive (Kaiser et al., 2024a). Learned surrogates replace the numerical update with graph networks over particles or mesh nodes (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021), neural operators (Li et al., 2021a), or physics-informed residuals (Raissi et al., 2019), and Hamiltonian and symplectic networks (Greydanus et al., 2019; Chen et al., 2020; Jin et al., 2020) constrain the evolution to be symplectic, as beam transport maps are (Wiedemann,

2015). However, these learned surrogates keep the full discretization or a fixed canonical coordinate as the state, so cost scales with its resolution.

Reduced-order models advance a low-dimensional state instead. For instance, a nonlinear manifold learned by a convolutional autoencoder (Lee & Carlberg, 2020), recurrent (Vlachas et al., 2022), or Koopman-linear (Lusch et al., 2018). All train the encoder through reconstruction, so a decoder is needed even when only the latent trajectory matters, with competing terms to balance (Champion et al., 2019). The noise in the observed field enters the target, biasing linear methods and leaving autoencoders robust only under explicit input corruption, thus, entangling the decoder error with the learned dynamics (Dawson et al., 2016; Vincent et al., 2008). Joint-embedding predictive architectures (JEPA) instead match the embedding of a target (LeCun et al., 2022), as shown for images (Assran et al., 2023) and video (Bardes et al., 2024). Unlike autoencoders, no decoder enters training and unpredictable detail may be discarded; collapse is prevented by a moving-average target encoder (Grill et al., 2020) or variance–covariance regularization (Bardes et al., 2022). The objective favors slow, predictable features (Sobal et al., 2022), which could suppress fast, low-amplitude modes.

Direct studies of JEPA as a state-space model are few. The closest precedent, Ulmen et al. (2025), learns a neural-ODE (Chen et al., 2018) latent state-space model of a pendulum from images with contractive embeddings and Lipschitz-bounded transitions. Like Sobal et al. (2022), it concerns a system whose true state is a handful of variables. Whether a JEPA keeps its accuracy–cost advantage, learns the dynamics rather than the training distribution, and filters observation noise when the state is a high-dimensional field has not been studied.

3 Longitudinal Beam Dynamics

Accelerators maintain dense bunches of charged particles traveling at relativistic speeds by utilizing electromagnetic radio-frequency (RF) cavities. The energy kick from the RF field, combined with the energy dependence of the revolution period, produces a restoring force that acts as a confining "bucket" keeping the bunch stable. The macroscopic spatial evolution of this particle bunch is strictly governed by microscopic, canonical physical invariants. The longitudinal phase space tracks the particle density over two canonically conjugate coordinates: energy deviation ΔE and phase deviation $\Delta\phi$, measured relative to an ideal reference particle at each turn in the accelerator. In a conservative regime, with fixed machine settings and negligible synchrotron radiation, the evolution of these coordinates follows Hamiltonian mechanics. At the particle level, a one-turn map M acts on the coordinates $(\Delta\phi, \Delta E)$. The phase-space density is transported along its trajectories, inducing a map $F : \mathcal{X} \to \mathcal{X}$ on states such that $x_{t+1} = F(x_t)$, where the state x_t depicts the phase-space density. Here, we assume that the map is deterministic and autonomous, with no external control driving the system, so a single state x_t determines the entire future trajectory.

Because the longitudinal dynamics derive from a Hamiltonian, the one-turn map M is symplectic; in this two-dimensional phase plane this is equivalent to exact area preservation, $\det(J_M) = 1$, where J_M is the Jacobian of M. By Liouville's theorem, the phase-space density is therefore transported without compression or dilution, so the occupied phase-space area is an exact invariant of the motion. In addition, small-amplitude particles execute approximately harmonic synchrotron oscillations in this phase plane at the synchrotron tune Q_s (oscillations per turn), which is fixed analytically by the machine settings. At larger amplitudes the motion is nonlinear. Whether a learned model respects these dynamics rests primarily on its predictive network, which must reproduce the action of F from one turn to the next, and the latent representation which must retain the information that F acts on.

4 Methodology

We treat the problem as system identification in a learned latent space: from density frames of the bunch over successive turns, learn a compact latent state and a transition map whose rollouts, read out in physical units, reproduce the one-turn map F. In this problem formulation, we can exactly determine whether a given learned predictor respects the dynamics because F is deterministic, autonomous, and symplectic, with a synchrotron tune fixed by machine settings.

Problem Formulation. Let $x_t \in \mathcal{X}$ be the longitudinal phase-space density at turn t, evolving as $x_{t+1} = F(x_t)$ and observed through frames $o_t = h(x_t)$. We learn an encoder E_θ mapping a

window of w frames to a latent $z_t \in \mathbb{R}^D$, a predictor P_ϕ advancing z_t one turn. A k-turn forecast is $\hat{y}_{t+k} = R_\psi(P_\phi^k(E_\theta(o_{t-w+1:t})))$, scored by the rollout error.

$$\mathcal{E}_K = \frac{1}{K}\sum_{k=1}^{K} \left\| \hat{y}_{t+k} - g(F^k(x_t)) \right\| \tag{1}$$

Within this setting, we address three research questions to evaluate the efficacy of JEPA for beam dynamics modeling. *(1) Physics constraints:* Does enforcing physics constraints on P_ϕ reduce $\mathcal{E}_K$ and improve long-horizon prediction relative to an unconstrained JEPA? *(2) Architecture:* At matched latent size D and compute, how does the JEPA compare with autoencoder-style (reconstruction-trained) and direct predictive (pixel-space) surrogates in $\mathcal{E}_K$ and training and inference times? *(3) Robustness:* For a corrupted initial state $\tilde{o}_0 = o_0 + \epsilon$, with ϵ drawn from Gaussian ($\mathcal{N}(0, \sigma^2 I)$) or Poisson (shot) noise, how does $\mathcal{E}_K$ grow with noise level, and does E_θ discard ϵ?

Baseline JEPA and Prediction Architectures. A Joint-Embedding Predictive Architecture (JEPA) comprises an encoder E_θ mapping observations to a latent state $z_t = E_\theta(o_t) \in \mathbb{R}^D$ and an autoregressive predictor P_ϕ approximating the transition $\hat{z}_{t+1} = P_\phi(z_t)$. Our BaselineJEPA instantiates this with a convolutional encoder that maps a single frame to a 64-dimensional latent and a two-hidden-layer MLP predictor. Rather than reconstructing pixels, it is trained in embedding space with a latent-invariance term and an anti-collapse regulariser, VISReg in our case. Both the latent size and the regulariser are ablated in Appendix C, where they emerge as the balance point between accuracy and stability.

Five variants replace the predictor with a volume-preserving map, $|\det J_P| = 1$, a property the true one-turn map has as a Hamiltonian flow and an unconstrained MLP has no reason to acquire. Four of them go further and partition the latent state into conjugate halves to impose full symplecticity. *Hamiltonian Neural Networks* (Greydanus et al., 2019) learn a general scalar Hamiltonian and integrate it. *Symplectic Recurrent Neural Networks* (Chen et al., 2020) restrict that Hamiltonian to a separable form, which makes an explicit symplectic step available. *SympNets* (Jin et al., 2020) compose symplectic layers directly, with no Hamiltonian and no integrator. We use two variants: *LA*, which alternates linear symplectic modules with elementwise nonlinear shears, and *G*, whose shears are gradients of a one-hidden-layer scalar network and which carries the universal approximation guarantee for symplectic maps. A fifth variant, *Linear-orth*, follows Koopman autoencoders, which advance the latent state with a single matrix (Lusch et al., 2018) that is orthogonal for a measure-preserving flow (Zhang et al., 2024).

To benchmark the JEPA against autoencoder-style architectures, we retrain the predictors SymptNet-LA and Linear-orth predictors under a reconstruction objective (Lusch et al., 2018), attaching the decoder while keeping the encoder/decoder configuration same. To benchmark against a purely predictive architecture, we add a Fourier Neural Operator (FNO) (Li et al., 2021a), which learns the one-turn map as a sequence of multiplications in the Fourier domain, so that each layer couples the whole grid. Every model is scored against a trivial baseline in the space its error is measured in. In observation space, *persistence* holds the last observed frame, $\hat{o}_k = o_0$ for every k. In latent space, the analogue holds the encoded state still, $\hat{z}_k = z_0$, scored against the encoding $E_\theta(o_k)$ of the true frame. The error in these trivial baselines grows as the bunch moves away from where it started. For implementation details of all architectures, please refer to Appendix A.

5 Experimental Setup

We first train the baseline JEPA and ablate its predictor, swapping the MLP for the five structure-preserving architectures with encoder, objective, data and window geometry fixed. Next, variants with least rollout error are carried forward to be compared against autoencoder and predictive style architectures on forecasting error, end-to-end cost, and robustness to observation noise and OOD.

FNAL Recycler. The Fermilab Recycler is a proton storage ring with a circumference of approximately 3.3 km, sharing a tunnel with the Main Injector. It operates at a nominal kinetic energy of 8 GeV and accumulates beam before transfer to the Main Injector for acceleration (Jackson, 1996; Ainsworth et al., 2020). In this work, we model Recycler longitudinal dynamics using BLonD (Beam Longitudinal Dynamics), a particle-tracking simulation suite developed at CERN (Timko

et al., 2023). These simulations provide the dynamical trajectories used to study JEPA-based prediction of longitudinal beam evolution. Appendix B provides the machine parameters and RF settings used in the simulations in detail.

Data Collection & Preprocessing. BLonD simulations consist of two datasets that differ only in the shape of the injected bunch: Gaussian and Non-Gaussian. A Gaussian bunch is fully described by its centroid and covariance. The non-Gaussian bunch places its particles along a spiral arm whose starting radius, pitch, width and radial envelope are drawn per run, with the same RF voltage, bunch length, energy spread and injection offset distributions as the Gaussian runs. This type of distribution is studied in order to study realistic filamented distributions coming from the downstream machines of the accelerator complex. The *Gaussian dataset* consists of $23{,}101$ runs; training is one turn in, one turn out (stride 1), 249 windows per run. A fixed seeded $70/15/15$ split gives $16{,}171$ training runs and $3{,}465$ each for validation and test ($4{,}026{,}579$ and $862{,}785$ windows), so held-out runs are complete simulations at unseen settings. The *non-Gaussian dataset* comprises of 216 runs ($53{,}784$ windows) and generated for OOD testing. Each run is 250 turns, each turn a 75×75 grid of macro-particle counts over $(\Delta\phi, \Delta E)$, with 10^5 macro-particles per run. A 250-turn record covers 0.88 of one nominal synchrotron period.

Training Procedure. Training uses AdamW (lr 3×10^{-3} at batch size 8192, weight decay 10^{-5}, gradient clipping at 1.0, plateau LR scheduler) for up to 500 epochs with early stopping on validation observation-space forecast error, patience 15. The reported checkpoint is the one that minimises that metric. Every run trains on a single GPU, an NVIDIA H100 (80 GB), in bfloat16 mixed precision.

Evaluation Metrics. We evaluate on three metrics classes. *(1) Latent geometry:* For each JEPA predictor we test the one-turn map through its Jacobian J_P: the per-dimension volume factor $|\det J_P|^{1/D}$, also composed over k turns, against the Liouville value 1; the singular-value pair products $\sigma_i \sigma_{D+1-i}$ against 1; and the defect $\|J_P^\top \Omega^* J_P - \Omega^*\|$ of the best-fitting skew-symmetric form Ω^*, together with the rank of Ω^*, which is D for a map that pairs every latent dimension. *(2) Forecast accuracy:* Every model rolls out true frame at turn 0 autoregressively to turn 249 on held-out simulation runs. Latent forecast error is the MSE between the free rollout $P_\phi^k(z_0)$ and the encoding $E_\theta(o_k)$ of the frame in observation space. We report MSE at turn 1, at turn 249 and average over the rollout. MSE on raw counts is dominated by the empty grid, so at turn 249 we add three beam-focused metrics: MSE on the beam core (pixels where the true frame exceeds 10% of its peak); centroid error, the distance between the predicted and true centres of mass; and mass ratio, the total predicted count over the total true count. For performance benchmarking of JEPA predictors against non-JEPA predictors, MSEs are reported as ratios against the persistence baseline. *(3) Computational cost:* Epochs trained until early stopping, training minutes per epoch, and the wall-clock time to roll out every held-out simulation for 249 turns.

6 RESULTS

Enforcing System Dynamics on Predictors. A bunch circulating in a storage ring can be approximated as a Hamiltonian system wherein every particle's position is paired with a conjugate momentum, and both states evolve under a single energy function. Between two frames the *one-turn map* carries each particle once around the ring, and Hamiltonian dynamics constrains that map. It is *symplectic*, preserving a skew form Ω that fixes which coordinate is conjugate to which (reciprocity). In the two-dimensional longitudinal plane it conserves phase-space volume (i.e., Liouville's theorem). Whether a learned one-turn predictor benefits from these constraints is an empirical question.

Figure 2 shows that BaselineJEPA fails all three properties of a symplectic map: volume preservation, singular-value reciprocity, and the existence of a preserved symplectic form. First, its median $\log_{10}|\det J_P| = -0.360$ gives a per-dimension Liouville factor $|\det J_P|^{1/D} = 0.9871$ (Figure 2b): the predictor discards 1.29% of the latent volume every turn. The baseline also breaks the reciprocity a symplectic map requires, $\sigma_i \cdot \sigma_{D+1-i} = 1$, with a median pair product off unity by 1.0×10^{-2} (Figure 2c), and the form Ω^* minimising the *symplectic defect* $\|J_P^T \Omega^* J_P - \Omega^*\|$ (Figure 2d), solved over the whole latent, pairs only 12 of its 64 dimensions. Every other predictor tested in the present study passes all three properties of a symplectic map. The four symplectic predictors (HNN, SRNN, SympNet-LA, and SympNet-G) are symplectic at initialisation with respect to the same declared form, pairing dimension i with $i + 32$; their nonzero defect in Figure 2d comes from restricting the step to a principal subspace it does not leave invariant. Linear-orth passes degenerately: $\det K = 1$,

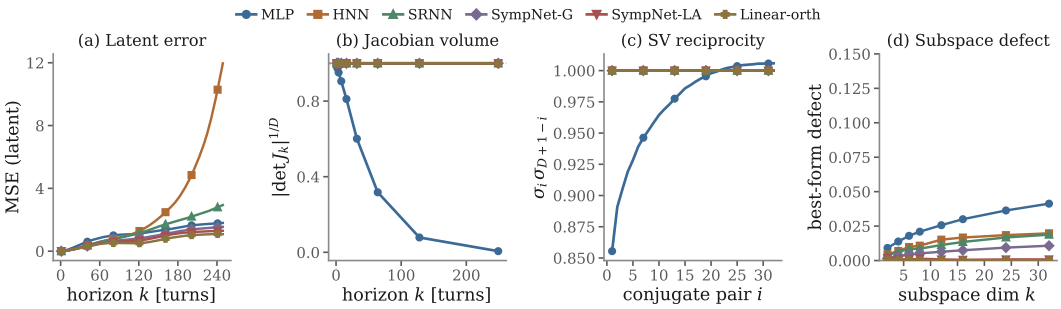

Figure 2: Performance benchmarking of JEPA predictors: (a) free-rollout latent MSE, (b) composed-Jacobian volume, (c) singular-value reciprocity, (d) best-form defect vs subspace dimension.

Table 1: Performance benchmarking of two predictors under the JEPA and Autoencoder objectives, with FNO for reference, on 3,465 held-out simulations. MSEs (including core MSE) depict ratios against a persistence baseline (hold the turn-0 frame fixed at every horizon).

		Training		Infer.	MSE ratio			Performance at t-249		
Predictor	Objective	epochs	min/ep.	rollout (s)	t-1	t-249	mean	core MSE	centroid	mass
persistence		–	–	–	7.20	759.5	1,747	1.24e4	0.014	1.00
SympNet-LA	JEPA‡	73	**3.90**	**5.03**	5.60	0.94	0.22	1.23	0.014	0.92
	Autoencoder	137	4.33	5.53	3.88	1.63e15	1.76e13	6.30e14	0.112	8.01e6
Linear-orth	JEPA‡	83	4.63	5.95	5.69	0.69	0.16	0.89	0.013	0.93
	Autoencoder	141	6.74	7.32	3.98	**0.31**	**0.07**	**0.37**	**0.006**	**0.99**
FNO†	–	98	14.03	28.36	**0.56**	25.40	5.23	15.13	0.107	1.11

†Batch size 1,024 (others 8,192).
‡The JEPA training numbers include the time to train the detached decoder and error numbers include error from decoder trained separately to bring prediction back to the observation space.

every singular value equals 1, and K decomposes into 2×2 rotations that exactly preserve a rank-64 block-diagonal Ω, though not the declared form.

Enforcing geometry on the predictor does not by itself buy accuracy (Figure 2a). HNN and SRNN predict worse than the baseline because both learn a scalar energy function and integrate Hamilton's equations, which is only correct if the latent coordinates are canonical position–momentum pairs, and nothing in the encoder's objective made them so. SympNet-G and SympNet-LA compose symplectic layers directly, assume nothing about which coordinate is conjugate to which, and beat the baseline. Linear-orth beats both SympNets. Orthogonal linear step is itself a Hamiltonian flow (quadratic H) and a sub-model of SympNet-LA, and inability to beat its own linear special case shows that SympNets are unable to leverage their nonlinearity, especially for one-turn step.

JEPA Performance Benchmarking. To gauge performance gains using JEPA, Table 1 compares, SympNet-LA and Linear-orth, predictors with the lowest free-rollout latent error, with their non-JEPA counterparts, retrained with the same encoder and decoder under the Koopman-autoencoder objective (forecast error + reconstruction error + latent residual, decoder attached). We name each version by predictor and objective, e.g. SympNet-LA (JEPA) and SympNet-LA (Autoencoder). Every model receives the true frame at turn 0 and rolls out 249 turns on all 3,465 held-out simulations.

JEPA trains $2.1\times$ (SympNet-LA) and $2.5\times$ (Linear-orth) faster and rolls out $1.1\times$–$1.2\times$ faster (Table 1). On autoregressive prediction accuracy though, Linear-orth (Autoencoder) outperforms every other model. FNO is the most accurate at a single turn, but its rollout diverges under its own predictions. SympNet-LA (Autoencoder) fails at long horizon while SympNet-LA (JEPA) stays bounded. The difference is what each predictor preserves: an orthogonal matrix preserves the length of the latent vector, so its rollout cannot leave a sphere, whereas a symplectic map preserves only volume and permits unbounded stretching along one direction if another is compressed by the same factor.

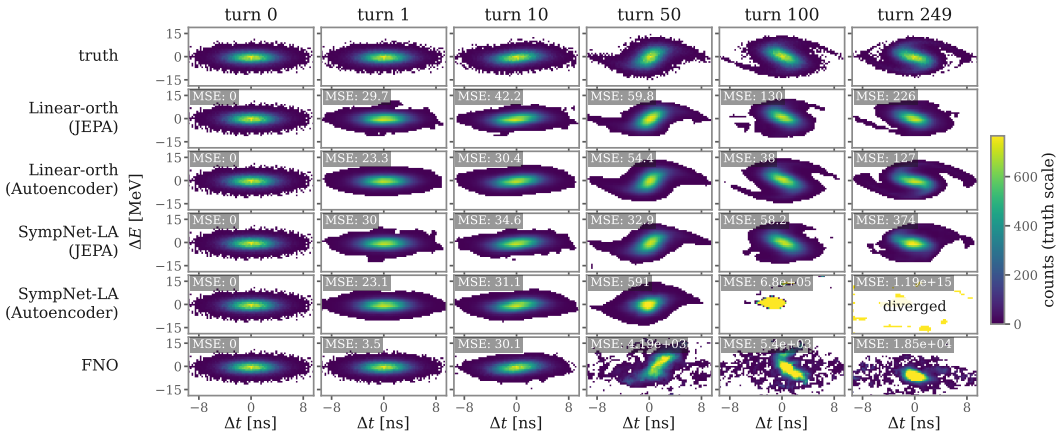

Figure 3: Autoregressive rollout of one held-out simulation for the five predictors against ground truth (top row). Columns are absolute turns; *MSE:* is each frame's error over the whole grid.

Linear-orth (Autoencoder) leads on core MSE as well: at turn 249 only it follows the curved filamentation arms of the true bunch, while the others leave density outside the true footprint, matching their mass ratios drifting from one (Figure 3). Centroid error measures show that every model stays centered except FNO and SympNet-LA (Autoencoder), which have already diverged. The persistence baseline (holding the turn-0 frame fixed) in Table 1 shows that the bunch's centroid moves over the rollout by 0.014. Interestingly, this error is quite close to both JEPAs' own centroid error. On the contrary, Linear-orth (Autoencoder) sits closer to the true centroid. This indicates that gradient from the decoded pixel error rewards tracking every visible feature of the bunch, including where it sits, while for JEPA objective ignores the small, slowly varying shift of the whole bunch, and therefore, neither the encoder nor the predictor is pressed to carry it through the rollout.

One caveat to note here is that all error metrics are reported in observation space, which favors the non-JEPA predictors. JEPA predictors are paired with a decoder trained on frozen latents, and their reported error includes the decoder's error, which no gradient was ever able to reduce through the encoder or predictor. At the first predicted turn, SympNet-LA (JEPA) and Linear-orth (JEPA) reach an MSE of 40.3 and 41.0, about $1.4\times$ their autoencoder partners (27.9 and 28.7), while in latent space their one-turn error is 21.5% and 20.1% of a persistence baseline.

Impact of Random Noise and OOD on System Performance. A single noisy measured frame can derail a long-horizon rollout. We corrupt the frame at turn 0 and roll out the models for 249 turns on all $3{,}465$ held-out simulations. Each noise level uses one seeded draw shared by all models. We use two noise families: Gaussian and Poisson. In this section, we will only discussion gaussian noise, results on Poisson distribution are discussed in Appendix D.

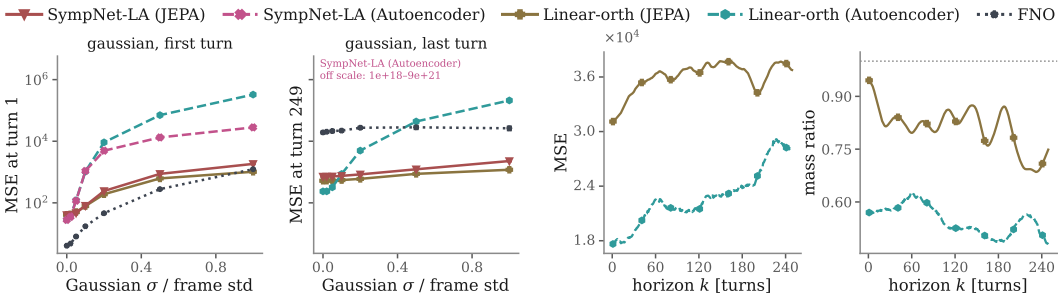

Figure 4: Left: Robustness of predictive arms to Gaussian noise in the starting frame, showing MSE at the first and last turn against noise level. Right: Linear-orth JEPA and Autoencoder rolled out on OOD non-Gaussian bunches, showing MSE and mass ratio against horizon.

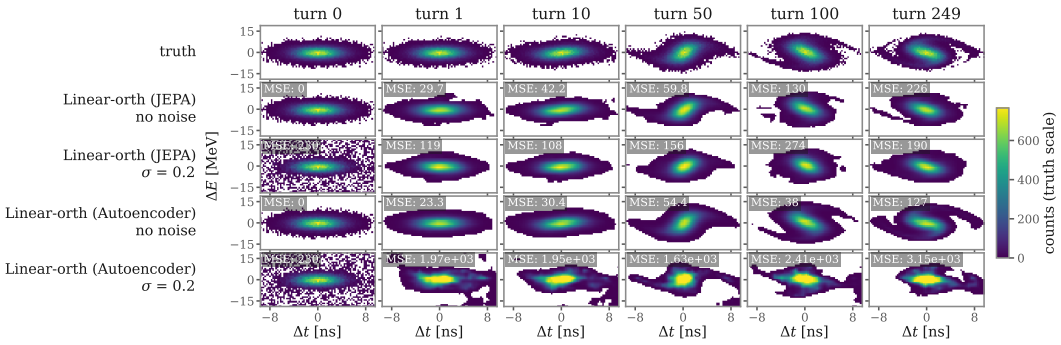

Figure 5: Phase-space rollout for Linear-orth under the JEPA and Autoencoder objectives, $\sigma = 0$ vs. $\sigma = 0.2$ Gaussian noise on the starting frame, against ground truth (top row).

Under Gaussian noise the autoencoder predictors break down at the first turn while JEPA predictors do not (Figure 4, first panel). The Gaussian family adds unclipped white noise at $\sigma \in \{0, 0.02, 0.05, 0.1, 0.2, 0.5, 1\}$, so it reaches every bin, including the empty background around the bunch. MSE at turn 1 measures the immediate damage. At $\sigma = 0.02$ the autoencoder error rises by 25% and 28%, the JEPA error by 2% and 3%. From $\sigma = 0.2$ upward the JEPA arms predict turn 1 with less error than the noisy frame they were given. The encoder filters part of the noise, while the autoencoder arms amplify it. FNO has the lowest turn-1 error at every level.

Over the full rollout the damage compounds for the autoencoder but not for JEPA (second panel). Without noise, Linear-orth (Autoencoder) is the most accurate model at turn 249; the order reverses between $\sigma = 0.05$ (319 against 532) and $\sigma = 0.1$ (865 against 549). SympNet-LA (Autoencoder) and FNO have already diverged from the uncorrupted initial condition, so noise does not change their standing. Figure 5 shows the two failure modes side by side at $\sigma = 0.2$: Linear-orth (JEPA) recovers a bunch close to the noise-free rollout, while Linear-orth (Autoencoder) draws a saturated core with spurious structure scattered outside the bunch. The accuracy advantage the autoencoder holds on noise-free inputs is therefore fragile: once the input is perturbed, the JEPA predictors are more robust, both at the first turn and across the full rollout.

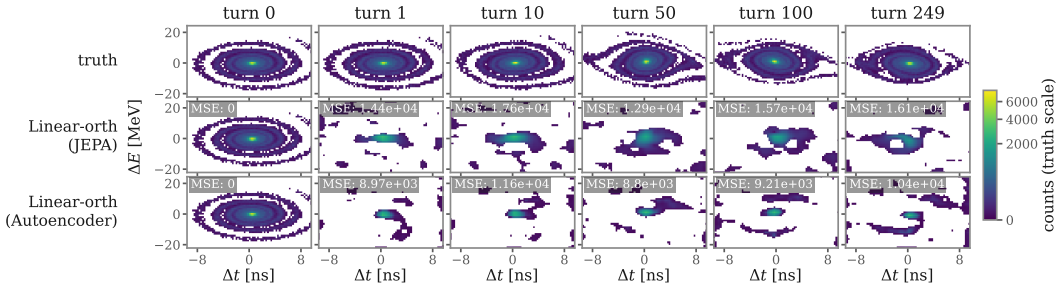

Figure 6: Neither Linear-orth arm reproduces a non-Gaussian bunch: autoregressive rollout of one held-out non-Gaussian simulation against ground truth (top row).

To gauge the efficacy of both architectures on OOD, we rolled out Linear-orth (JEPA) and Linear-orth (Autoencoder), trained on Gaussian bunches, on 216 non-Gaussian simulations without retraining. While Figure 4 (right) suggests the JEPA predictor generalises better, Figure 6 reveals that from the first predicted turn neither arm draws the spiral arms and both collapse the bunch.

7 Discussion

A latent world model learns two components jointly on one objective. The predictor encodes the dynamics and the encoder decides what the predictor acts on. The results in the previous section show that while JEPA beats non-JEPA architectures on training and robustness to noise, their autoencoder counterparts beats them over long horizon forecasts. The performance gap depicts the JEPA latent

failing to carry the higher-order moments of the density which is further validated by its inability to perform on OODs. The rest of this section reads those results against what is already known about learning representations of dynamical systems.

We noticed previously that, while JEPA architectures match the persistence baseline on centroid error, the autoencoders tracks it. The latent is predictable and incomplete at once. This is a known failure of latent-regression objectives. Kostic et al. (2022; 2024) show that minimising forecast error in latent space admits degenerate minimisers, functions that are invariant because they discard the state, and that JEPA's regularisers only prevent the latent from becoming constant. Their DPNets drop the regression loss and instead score the encoder on how invariant its span is, penalise metric distortion so the map stays injective, and fit the operator afterwards by least squares. VAMPnets' variational bound (Mardt et al., 2018) and the reconstruction term in time-lagged autoencoders (Wehmeyer & Noé, 2018) serve the same role. The JEPA speed and noise advantages we observe are consistent with this view: the JEPA target is a conditional expectation, so zero-mean noise averages out rather than being reconstructed. The open problem is to keep that property while adopting an invariance-based objective, which no current JEPA does to the best of our knowledge.

SympNets, HNN, and SRNN pass every geometric test, yet only SympNets beat the MLP. The difference is what each predictor demands of the encoder. Symplectic layers constrain only the step, whereas a Hamiltonian integrator also assumes the latent splits into conjugate position and momentum, and nothing in the objective asks the encoder to supply them. SympNets compose symplectic shears on the same halves but never require them to be canonical. Linear-orth goes further: its single orthogonal step, borrowed from the Koopman autoencoder, preserves norm and volume by construction, and beats both SympNets. Botev et al. (2021) and Toth et al. (2019) report the same fragility when Hamiltonian priors move from state space to pixels. Canonical latents have been learned by construction for point-valued states, through symplectic flows to action–angle or Darboux coordinates (Li et al., 2020; Jin et al., 2022) and through symplectic autoencoders for reduced Hamiltonian models, including Vlasov via particle discretisation (Buchfink et al., 2023; Tyranowski & Kraus, 2023). However, the state here is a *density*, whose evolution under the Liouville operator is linear, and to the best of our knowledge, none learns canonical latents from density frames.

To be truly a world model for system dynamics, JEPA architectures must perform well on OOD that reflect real-world noise and data/concept-drifts. In this case, the Gaussian is fixed by its first two moments, so the latent never had to carry more, and no loss can supply moments the data never exercised. One potential fix is to impose linearity. The Liouville operator is linear on densities, so a non-Gaussian bunch is a superposition of Gaussian ones and evolves as that superposition. A linear latent step though alone does not inherit this, as Linear-orth shows. The encoder must also be linear in the density, for instance as expectations of a feature dictionary, as in extended dynamic mode decomposition (Williams et al., 2015). While linearity can resolve the unseen moments challenges, it still cannot fully resolve the gap in training data, such as spiral arms in training bunch. Closing this gap calls for complementary measures: objectives that learn the invariant subspaces of the transfer operator rather than the training trajectories (Kostic et al., 2024), training across environments that separates shared from setting-specific dynamics (Yin et al., 2021; Kirchmeyer et al., 2022), pretraining on broader families of initial states (McCabe et al., 2024), and correcting the latent state with measurements during deployment.

8 Conclusion

We studied the latent dynamics of a JEPA world model for longitudinal beam dynamics and found symplectic predictors restore unit volume, reciprocal singular values and full-rank pairing, but geometry alone does not order accuracy. This is show in the HNN and SRNN examples which predict worse than the baseline, even though these methods provide a learned Hamiltonian to an integrator and thereby assume the encoder's coordinates are canonical. When benchmarked against non-JEPA counterparts, we observed faster training ($2.1\times$–$2.5\times$) and inference ($1.1\times$–$1.2\times$) with JEPA. The reconstruction objective yields the lowest forecast error in the absence of noise, however, when Gaussian noise is introduced in the initial frame, the JEPA encoder filters the perturbation while the autoencoder amplifies it. These findings align with prior work on learning transfer operators and latent Hamiltonian dynamics. Future work should therefore focus on encoders built for phase-space densities that learn canonical, density-linear latents under an invariance-based objective.

AI USE STATEMENT

We acknowledge the use of generative AI tools accessed through the [REDACTED] AI Incubator (LINK REDACTED): Claude Opus 5.5, Claude Sonnet 5, and Claude Fable 5.1 from Anthropic, and GPT Astra from OpenAI. These tools were used to brainstorm research directions, to suggest candidate literature, to write and debug code for the model architectures, and to draft and edit parts of the manuscript. The authors manually reviewed and validated all AI-generated code, and revised all AI-assisted text for accuracy, relevance, and intended meaning. The authors read the suggested literature and selected the references cited in this work. All final content was reviewed and approved by the authors.

ETHICS STATEMENT

This work does not involve human subjects, personal data, or information about identifiable individuals. All data are produced by the open-source BLonD particle-tracking code using published machine parameters of the FNAL Recycler (Appendix B), so no privacy, consent, or data-licensing concerns arise. The intended application is forecasting and diagnosis in particle accelerator operations, a safety- and mission-critical setting. Our results show that the studied models fail on out-of-distribution bunch shapes and can diverge over long horizons. We therefore caution against deploying such surrogates as replacements for validated simulation codes or for machine-protection and interlock systems; they are better suited to supporting human operators, with outputs verified against established tools. Beyond that, we are not aware of harmful applications specific to this work. The authors declare no competing interests.

REPRODUCIBILITY STATEMENT

Section 5 describes the data generation, the fixed seeded 70/15/15 split, the training procedure and hyperparameters, the hardware, and all evaluation metrics. Appendix A specifies every model compared in this work, including encoder, decoder, and predictor architectures, integrator choices, and parameter counts (Table 2). Appendix B lists the machine constants and RF settings used in the BLonD simulations (Table 3), and Appendix C reports the regulariser and latent-size ablations that fix the default configuration. The noise protocols, including levels and seeded draws shared across models, are given in Section 6 and Appendix D. Code and data could be made available from the authors upon reasonable request.

REFERENCES

R. Ainsworth, P. Adamson, J. Amundson, I. Kourbanis, Q. Lu, and E. Stern. High intensity space charge effects on slip stacked beam in the Fermilab Recycler. *Phys. Rev. Accel. Beams*, 22: 020404, 2 2019. doi: 10.1103/PhysRevAccelBeams.22.020404. URL https://link.aps.org/doi/10.1103/PhysRevAccelBeams.22.020404.

R. Ainsworth, Philip Adamson, Bruce C. Brown, David Capista, Kyle Hazelwood, Ioanis Kourbanis, Denton K Morris, Meiqin Xiao, and Ming-Jen Yang. High intensity operation using proton stacking in the Fermilab Recycler to deliver 700 kw of 120 gev proton beam. *Phys. Rev. Accel. Beams*, 23:121002, 12 2020. doi: 10.1103/PhysRevAccelBeams.23.121002. URL https://link.aps.org/doi/10.1103/PhysRevAccelBeams.23.121002.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Mahmoud Assran et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning, 2025.

Omri Azencot, N. Benjamin Erichson, Vanessa Lin, and Michael W. Mahoney. Forecasting sequential data using consistent koopman autoencoders. In *International Conference on Machine Learning (ICML)*, pp. 475–485, 2020.

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

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. In *European Conference on Computer Vision*, 2024.

Aleksandar Botev, Andrew Jaegle, Peter Wirnsberger, Daniel Hennes, and Irina Higgins. Which priors matter? benchmarking models for learning latent dynamics. *arXiv preprint arXiv:2111.05458*, 2021.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022.

Patrick Buchfink, Silke Glas, and Bernard Haasdonk. Symplectic model reduction of hamiltonian systems on nonlinear manifolds and approximation with weakly symplectic autoencoder. *SIAM journal on scientific computing*, 45(2):A289–A311, 2023.

Kathleen Champion, Beth Lusch, J. Nathan Kutz, and Steven L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45): 22445–22451, 2019.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Zhengdao Chen, Jianyu Zhang, Martin Arjovsky, and Léon Bottou. Symplectic recurrent neural networks. In *International Conference on Learning Representations*, 2020.

Scott T. M. Dawson, Maziar S. Hemati, Matthew O. Williams, and Clarence W. Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57(3):42, 2016.

Quentin Garrido, Mahmoud Assran, Nicolas Ballas, Adrien Bardes, Laurent Najman, and Yann LeCun. Learning and leveraging world models in visual representation learning. *arXiv preprint arXiv:2403.00504*, 2024.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

G. Jackson. The Fermilab Recycler Ring technical design report. Revision 1.2. Technical Report FNAL-TM-1991ON: DE97051388; BR: KAHEP, Fermilab, 11 1996. URL `https://www.osti.gov/biblio/16029`.

Pengzhan Jin, Zhen Zhang, Aiqing Zhu, George Em Karniadakis, and Lu Lu. SympNets: Intrinsic structure-preserving symplectic networks for identifying hamiltonian systems. *Neural Networks*, 132, 2020.

Pengzhan Jin, Zhen Zhang, Ioannis G Kevrekidis, and George Em Karniadakis. Learning poisson systems and trajectories of autonomous systems via poisson neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8271–8283, 2022.

Jan Kaiser, Chenran Xu, Annika Eichler, and Andrea Santamaria Garcia. Bridging the gap between machine learning and particle accelerator physics with high-speed, differentiable simulations. *Physical review accelerators and beams*, 27(5):054601, 2024a.

Jan Kaiser, Chenran Xu, Annika Eichler, and Andrea Santamaria Garcia. Bridging the gap between machine learning and particle accelerator physics with high-speed, differentiable simulations. *Phys. Rev. Accel. Beams*, 27:054601, May 2024b. doi: 10.1103/PhysRevAccelBeams.27.054601. URL https://link.aps.org/doi/10.1103/PhysRevAccelBeams.27.054601.

Matthieu Kirchmeyer, Yuan Yin, Jérémie Donà, Nicolas Baskiotis, Alain Rakotomamonjy, and Patrick Gallinari. Generalizing to new physical systems via context-informed dynamics model. In *International conference on machine learning*, pp. 11283–11301. PMLR, 2022.

Vladimir Kostic, Pietro Novelli, Andreas Maurer, Carlo Ciliberto, Lorenzo Rosasco, and Massimiliano Pontil. Learning dynamical systems via koopman operator regression in reproducing kernel hilbert spaces. *Advances in Neural Information Processing Systems*, 35:4017–4031, 2022.

Vladimir Kostic, Pietro Novelli, Riccardo Grazzi, Karim Lounici, et al. Learning invariant representations of time-homogeneous stochastic dynamical systems. In *International Conference on Learning Representations*, volume 2024, pp. 54329–54341, 2024.

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

Kookjin Lee and Kevin T Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.

Shuo-Hui Li, Chen-Xiao Dong, Linfeng Zhang, and Lei Wang. Neural canonical transformation with symplectic flows. *Physical Review X*, 10(2):021020, 2020.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations (ICLR)*, 2021a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021b.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural PDE solvers. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021.

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

Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. Vampnets for deep learning of molecular kinetics. *Nature communications*, 9(1):5, 2018.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, et al. Multiple physics pretraining for spatiotemporal surrogate models. *Advances in Neural Information Processing Systems*, 37:119301–119335, 2024.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.

Ji Qiang, Robert D Ryne, Salman Habib, and Viktor Decyk. An object-oriented parallel particle-in-cell code for beam dynamics simulation in linear accelerators. In *Proceedings of the 1999 ACM/IEEE conference on Supercomputing*, pp. 55–es, 1999.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pp. 8459–8468. PMLR, 2020.

Vlad Sobal, Jyothir SV, Siddhartha Jalagam, Nicolas Carion, Kyunghyun Cho, and Yann LeCun. Joint embedding predictive architectures focus on slow features. *arXiv preprint arXiv:2211.10831*, 2022.

H. Timko, S. Albright, T. Argyropoulos, H. Damerau, K. Iliakis, L. Intelisano, B. E. Karlsen-Baeck, I. Karpov, A. Lasheen, L. Medina, D. Quartullo, J. Repond, A. L. Vanel, J. Esteban Müller, M. Schwarz, P. Tsapatsaris, and G. Typaldos. Beam longitudinal dynamics simulation studies. *Phys. Rev. Accel. Beams*, 26:114602, Nov 2023. doi: 10.1103/PhysRevAccelBeams.26.114602. URL https://link.aps.org/doi/10.1103/PhysRevAccelBeams.26.114602.

Helga Timko, Simon Albright, Theodoros Argyropoulos, Konstantinos Iliakis, Birk Emil Karlsen-Bæck, Ivan Karpov, Alexandre Lasheen, Luis Eduardo Medina, Danilo Quartullo, Joël Repond, et al. Beam longitudinal dynamics simulation suite blond. *arXiv preprint arXiv:2206.08148*, 2022.

Peter Toth, Danilo Jimenez Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. *arXiv preprint arXiv:1909.13789*, 2019.

Tomasz M Tyranowski and Michael Kraus. Symplectic model reduction methods for the vlasov equation. *Contributions to Plasma Physics*, 63(5-6):e202200046, 2023.

Jonas Ulmen, Ganesh Sundaram, and Daniel Görges. Learning state-space models of dynamic systems from arbitrary data using joint embedding predictive architectures. *IFAC-PapersOnLine*, 59(18):19–24, 2025.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103. ACM, 2008.

Pantelis R Vlachas, Georgios Arampatzis, Caroline Uhler, and Petros Koumoutsakos. Multiscale simulations of complex systems by learning their effective dynamics. *Nature Machine Intelligence*, 4(4):359–366, 2022.

Christoph Wehmeyer and Frank Noé. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *The Journal of chemical physics*, 148(24), 2018.

Helmut Wiedemann. *Particle Accelerator Physics*. Graduate Texts in Physics. Springer, 4 edition, 2015.

Steffen Wiewel, Moritz Becher, and Nils Thuerey. Latent space physics: Towards learning the temporal evolution of fluid flow. In *Computer graphics forum*, volume 38, pp. 71–82. Wiley Online Library, 2019.

Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

Yuan Yin, Ibrahim Ayed, Emmanuel de Bézenac, Nicolas Baskiotis, and Patrick Gallinari. Leads: Learning dynamical systems that generalize across environments. *Advances in Neural Information Processing Systems*, 34:7561–7573, 2021.

Jingdong Zhang, Qunxi Zhu, and Wei Lin. Learning hamiltonian neural koopman operator and simultaneously sustaining and discovering conservation laws. *Physical Review Research*, 6(1): L012031, 2024.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

Table 2: Implementation details of all models compared in this work. Every model consumes one warm-up turn and predicts one turn ahead during training.

Model	State	One-turn step	Objective	Parameters
Baseline JEPA	$D = 64$	MLP, $64 \to 128 \to 128 \to 64$	VISReg, invariance	1,642,369
with HNN	$D = 64$	implicit midpoint on $H_\theta(q, p)$	VISReg, invariance	1,634,241
with SRNN	$D = 64$	leapfrog on $T_\theta(p) + V_\theta(q)$	VISReg, invariance	1,651,009
with SympNet-LA	$D = 64$	linear and activation shears	VISReg, invariance	1,617,825
with SympNet-G	$D = 64$	gradient shears	VISReg, invariance	1,626,753
with Linear-orth	$D = 64$	orthogonal matrix	VISReg, invariance	1,613,377
Autoencoder Arms				
SympNet-LA	$D = 64$	linear and activation shears	forecast, recon., latent	1,617,825
Linear-orth	$D = 64$	orthogonal matrix	forecast, recon., latent	1,613,377
FNO	grid	three spectral blocks, 12 modes	forecast MSE	1,772,737

A Model Architectures and Implementation

Table 2 summarises models implemented for this study. All models consume one warm-up turn and predict one turn ahead during training, and all are trained under the procedure of Section 5.

A.1 The JEPA family

Encoder and decoder. We encode a single frame with three stride-2 convolutions of widths $32 \to 64 \to 128$, each followed by an ELU nonlinearity and batch normalisation, and project the flattened result linearly to $z_t \in \mathbb{R}^{64}$. We decode with the mirrored transposed-convolution stack and a softplus head, so decoded frames are non-negative rates, and resize bilinearly to the 75×75 grid. The encoder contains 912,384 parameters and the decoder 696,897.

The objective is defined entirely in latent space; the decoder is a read-out attached to it rather than a part of it. Its term compares the decoded frame against the observation whose *encoder* latent it was given,

$$\mathcal{L}_{\text{rec}} = \left\| D_\psi\big(\text{sg}\big[E_\theta(o_{t+1})\big]\big) - o_{t+1} \right\|^2, \tag{2}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient. Two consequences follow. The decoder is trained to invert the encoder and never the predictor, since its input is $E_\theta(o_{t+1})$ and not $P_\phi(E_\theta(o_t))$. Because that input is detached, no gradient of $\mathcal{L}_{\text{rec}}$ reaches the encoder or the predictor, so the decoder cannot shape the latent space it reads from. Observation-space forecasts are formed by decoding predicted latents, $\hat{o}_{t+k} = D_\psi(\hat{z}_{t+k})$, outside the gradient computation. The decoder's own reconstruction error is therefore a floor below which no observation-space forecast can fall, however good the latent prediction is.

Baseline predictor. We advance the latent with a two-hidden-layer MLP of widths $64 \to 128 \to 128 \to 64$. The predictor contains 33,088 parameters.

A.1.1 Structure-preserving predictors

Each variant treats the latent state as $s \equiv z_t \in \mathbb{R}^{64}$ and partitions it into conjugate halves $s = (q, p)$ with $q, p \in \mathbb{R}^{32}$, pairing latent dimension i with $i + 32$ and declaring the canonical symplectic form

$$\Omega = \begin{pmatrix} 0 & I_{32} \\ -I_{32} & 0 \end{pmatrix}. \tag{3}$$

A map P is symplectic when $J_P(s)^\top \Omega J_P(s) = \Omega$ at every state, where $J_P(s)$ is its Jacobian. Each variant advances the latent state by one time step and receives no conditioning input.

Hamiltonian Neural Networks (HNNs) (Greydanus et al., 2019) parameterize a scalar Hamiltonian and define the dynamics through Hamilton's equations. Of the four it makes the weakest structural commitment: H_θ is an arbitrary scalar function of the full latent state, given no separable or physical form, and symplecticity enters only through the integrator.

We model a general Hamiltonian $H_\theta(q, p)$ of the partitioned latent state using an unconditioned MLP with widths $64 \to 128 \to 128 \to 1$. Both hidden layers use $\tanh$ activations and biases; the output layer has no bias. The vector field $f_\theta(s) = (\nabla_p H_\theta, -\nabla_q H_\theta)$ is computed by automatic differentiation. We replace the RK4 integration used in the original HNN work with implicit midpoint at a step size of $h = 1$ turn and no substepping, because RK4 is not symplectic: its Jacobian does not satisfy $J^\top \Omega J = \Omega$. Implicit midpoint evaluates the vector field halfway between the current state and the next one, so the next state appears on both sides of its own update equation and must be solved for. We do not solve it to convergence. Each step instead runs a fixed budget of eight fixed-point iterations,

$$u^{(0)} = s_t, \qquad u^{(k+1)} = s_t + h\, f_\theta\left(\frac{s_t + u^{(k)}}{2}\right), \qquad k = 0, \ldots, 7, \tag{4}$$

with prediction $\hat{s}_{t+1} = u^{(8)}$, and gradients propagate through all iterations. The exactly solved midpoint rule is symplectic; the finite-iteration implementation does not guarantee exact symplecticity. The predictor contains $24{,}960$ parameters.

Symplectic Recurrent Neural Networks (SRNNs) (Chen et al., 2020) combine a learned separable Hamiltonian with a symplectic integrator to define a recurrent dynamics model. Separability is its distinguishing assumption: restricting H_θ to a sum of a p-dependent and a q-dependent term is what makes an explicit, exactly symplectic step available.

On the same conjugate halves, we parameterize

$$H_\theta(q, p) = T_\theta(p) + V_\theta(q). \tag{5}$$

The functions T_θ and V_θ are independent, unconditioned MLPs, each with widths $32 \to 128 \to 128 \to 1$. Both hidden layers use $\tanh$ activations and biases; the output layers have no bias. Hamiltonian gradients are computed by automatic differentiation. We use the explicit leapfrog (Störmer–Verlet) integrator with step size $h = 1$ turn and no substepping. Unlike the approximate implicit solve used for our HNN, this update is symplectic by construction, up to floating-point error, and requires no iterative solver. The predictor contains $41{,}728$ parameters.

SympNets (Jin et al., 2020) directly predict the next latent state by composing symplectic layers, without learning a Hamiltonian or using a numerical integrator. Both variants are built from *shears*: maps that leave one half of the latent state unchanged and add to the other half a function of that unchanged half, as in $(q, p) \mapsto (q, p + g(q))$. Because the increment depends only on the half being held fixed, a shear has a triangular Jacobian with unit diagonal, hence unit determinant, and the composition of shears is symplectic up to floating-point error.

SympNet-LA alternates linear modules with activation modules. We use four layers with shear width 128 and a sigmoid activation. The predictor contains $8{,}544$ parameters.

SympNet-G uses gradient modules, in which the shear increment is the gradient of a scalar function, so the shear matrix is a Hessian and is symmetric at any width. Capacity is therefore raised by widening the module rather than by adding layers. We use four layers with shear width 128 and a sigmoid activation. The predictor contains $17{,}472$ parameters.

Linear-orth takes its step from the Koopman autoencoder (Lusch et al., 2018), which advances the latent state by a single matrix, and constrains that matrix to be orthogonal, as a measure-preserving flow requires (Zhang et al., 2024). It does not partition the latent state and declares no symplectic form. The step is applied directly, with no integrator and no bias,

$$\hat{z}_{t+1} = Q_\theta\, z_t, \qquad Q_\theta = B \exp(A_\theta), \qquad A_\theta = L_\theta - L_\theta^\top, \tag{6}$$

where L_θ is the lower-triangular part of a free 64×64 parameter matrix, so A_θ is skew-symmetric and $\exp(A_\theta)$ is orthogonal, and B is a fixed orthogonal matrix obtained by orthogonalising (QR) the default initialisation of the layer; A_θ starts at zero, so $Q_\theta = B$ at initialisation. Every singular value of Q_θ is 1 and every eigenvalue lies on the unit circle, so the step preserves the length of the latent vector as well as its volume, and no latent direction grows or decays over a free rollout. Since $\det \exp(A_\theta) = 1$, $\det Q_\theta = \det B$ throughout training, so the step keeps the orientation it was initialised with. The predictor contains $4{,}096$ parameters, of which only the $2{,}016$ below the diagonal, the dimension of the rotation group $SO(64)$, affect Q_θ.

Table 3: Machine constants of the simulated Recycler. The lower block is derived from the upper.

Quantity	Symbol	Value
Circumference	C	3319.4188 m
Harmonic number	h	588
Synchronous momentum	p_s	8.835 GeV/c (protons)
Transition gamma	γ_t	20.2576
RF peak voltage, per station	V_a, V_b	60–100 kV, sampled independently
RF phase	ϕ_{rf}	π (stationary bucket, no ramp)
Synchronous energy	E_s	8.885 GeV ($\gamma = 9.469$, $\beta = 0.99441$)
Slip factor	η_0	-8.716×10^{-3}
Revolution frequency	f_{rev}	89.810 kHz ($T_{\mathrm{rev}} = 11.135\,\mu$s)
RF frequency	hf_{rev}	52.808 MHz
Phase slip per eV	d	-3.665×10^{-9} rad/eV
Synchrotron tune	Q_s	3.5×10^{-3} (period 284 turns; $\pm 5\,\%$ across runs)

A.2 Latent surrogates

Koopman autoencoder (Lusch et al., 2018). We encode one frame to $D = 64$ with convolution widths $32 \to 64 \to 128$, and decode with the mirrored transposed-convolution stack under a softplus head, dividing the input by 100 and multiplying the output by the same factor. We advance the code by a single linear map K_θ with no bias, so the step has no fixed point at which it can place the bunch. We train on the sum of the forecast error, the reconstruction error, and the latent residual

$$\mathcal{L}_{\mathrm{lin}} = \left\| K_\theta\, z_t - E_\theta(o_{t+1}) \right\|^2, \tag{7}$$

each at unit weight, the last of which requires the map to advance the encoder's own state rather than merely produce a decodable one. *Linear-orth (Autoencoder)* parameterises K_θ as orthogonal, so every eigenvalue lies on the unit circle and $K_\theta^{-1} = K_\theta^\top$ up to floating-point error. *SympNet-LA (Autoencoder)* uses the SympNet-LA predictor of Section A.1 as K_θ; that map is nonlinear, so this variant is a Koopman autoencoder in its objective only. We use no backward-consistency term (Azencot et al., 2020) in either variant. The variants contain 1,613,377 and 1,617,825 parameters, the same as their JEPA counterparts.

B FNAL Recycler and Simulation Settings

The Fermilab Recycler is a proton storage ring with a circumference of approximately 3.3 km, sharing a tunnel with the Main Injector. It operates at a nominal kinetic energy of 8 GeV and accumulates beam before transfer to the Main Injector for acceleration (Jackson, 1996; Ainsworth et al., 2020). A central longitudinal manipulation is *slip-stacking*, in which two radio-frequency (RF) systems operating at slightly different frequencies confine two sets of proton bunches at different momenta. The bunch trains consequently drift longitudinally relative to one another, allowing them to be aligned for subsequent combination and approximately doubling the intensity per bunch delivered for acceleration (Ainsworth et al., 2019). Accurate modeling of these dynamics supports the design of RF manipulations that preserve beam quality and limit particle losses.

In this work, we model Recycler longitudinal dynamics using BLonD (Beam Longitudinal Dynamics), a particle-tracking simulation suite developed at CERN (Timko et al., 2023). The simulated reference momentum is $p_s = 8.835$ GeV/c, giving a revolution frequency of approximately 89.810 kHz. With harmonic number $h = 588$, the corresponding reference RF frequency is 52.808 MHz. Table 3 summarizes the machine parameters and RF settings used in the simulations. These simulations provide the dynamical trajectories used to study JEPA-based prediction of longitudinal beam evolution.

C Ablation Studies

Two ablations test were conducted on Linear-orth (JEPA) arm: regulariser and latent width. Each changes one setting of that arm ($D = 64$, VISReg) and keeps the encoder and decoder widths,

the orthogonal predictor, the latent-invariance term, the data and the training budget. Every run is trained once and scored exactly as Table 1: a free 249-turn rollout from the true turn-0 frame on the same 3,465 held-out simulations, with each error reported as a ratio to persistence at the same horizon, so a ratio below 1 beats holding the first frame still.

Table 4: Loss-function ablation of Linear-orth (JEPA). MSEs (including core MSE) depict ratios against a persistence baseline (hold the turn-0 frame fixed at every horizon).

	MSE ratio			Performance at t-249		
Regulariser	t-1	t-249	mean	core MSE	centroid	mass
persistence[§]	7.20	759.5	1,747	1.24e4	0.014	1.00
VISReg	5.69	0.69	0.16	0.89	0.013	0.93
SIGReg	11.02	1.47	0.60	1.87	0.019	0.86
VICReg	**4.88**	**0.33**	**0.08**	**0.40**	**0.006**	**0.99**
variance only	5.68	3.25	0.58	2.67	0.023	1.13

[§]Absolute values, not ratios; multiply any ratio above by this row to recover its raw value.

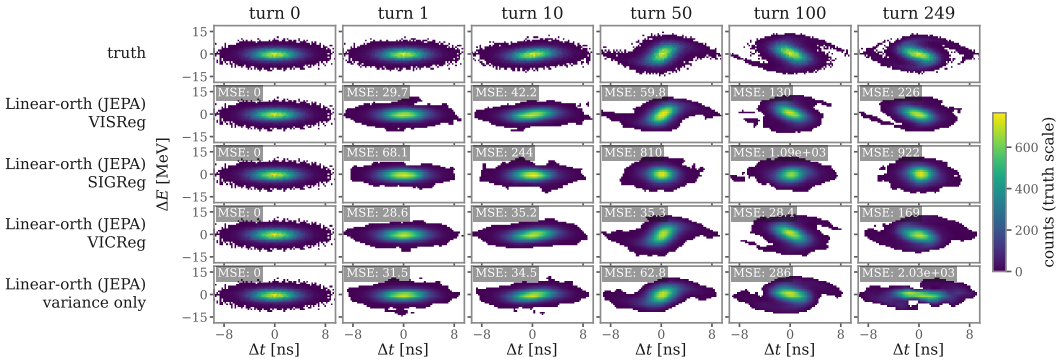

Figure 7: Autoregressive rollout of Linear-orth (JEPA) under each anti-collapse regulariser on the held-out simulation of Figure 3, against ground truth (top row). Columns are absolute turns; *MSE:* is each frame's error over the whole grid.

C.1 Loss Metrics

We replace VISReg with three alternatives and keep the latent-invariance term. VISReg matches each batch's per-dimension scale, centre and sliced marginal shape to a standard Gaussian. SIGReg (Balestriero & LeCun, 2025) tests the raw latent against $\mathcal{N}(0, I)$ with a single goodness-of-fit statistic on random one-dimensional projections. VICReg (Bardes et al., 2022), at its published weights, keeps each dimension's standard deviation above a floor and penalises covariance between dimensions. Variance only keeps the floor alone, the least that prevents collapse. *If any non-collapsed latent served the predictor equally well, the four would score alike; if the regulariser shapes the latent that the predictor rolls forward, they would separate at long horizons.*

They separate at long horizons (Table 4). VICReg is best on every column: it halves VISReg's mean ratio (0.080 vs 0.159) and core MSE ratio (0.40 vs 0.89), and conserves mass to within 1% (0.99 vs 0.93). VICReg and variance only differ only in the covariance term; without it the turn-249 ratio rises tenfold, from 0.33 to 3.25, and the rollout gains 13% in mass, so it is the decorrelation, not the variance floor, that keeps the rollout stable. On the simulation of Figure 3, the SIGReg rollout barely rotates the bunch and the variance-only rollout loses its shape by turn 249 (Figure 7).

VICReg's gain rests on its covariance term at its published weights: three coefficients tuned on natural images, a floor that fixes the scale rather than the distribution, and an $O(D^2)$ penalty whose balance against the invariance term shifts with D and batch size, as the tenfold jump to variance only shows. VISReg addresses these by matching the centre, scale and sliced marginal shape to a single fixed target, $\mathcal{N}(0, I)$, with no per-term weights, and still yields a stable rollout that conserves

mass to 7% and beats persistence at turn 249 (0.69). The natural next step, SIGReg's full goodness-of-fit test against the same target, does not carry over to this dataset and barely rotates the bunch in Figure 7. VISReg is therefore the point between VICReg's tuned-weight lead and SIGReg's failure that we adopt throughout.

Table 5: Latent-dimension ablation of Linear-orth (JEPA). MSEs (including core MSE) depict ratios against a persistence baseline (hold the turn-0 frame fixed at every horizon).

	MSE ratio			Performance at t-249		
Latent D	t-1	t-249	mean	core MSE	centroid	mass
persistence[§]	7.20	759.5	1,747	1.24e4	0.014	1.00
16	7.93	0.75	0.27	0.95	0.012	**0.96**
32	7.27	1.26	0.31	1.56	0.017	0.94
64	5.69	0.69	0.16	0.89	0.013	0.93
96	5.25	**0.62**	0.15	**0.81**	**0.010**	0.92
128	**5.09**	0.63	**0.14**	0.83	0.011	0.91

[§]Absolute values, not ratios; multiply any ratio above by this row to recover its raw value.

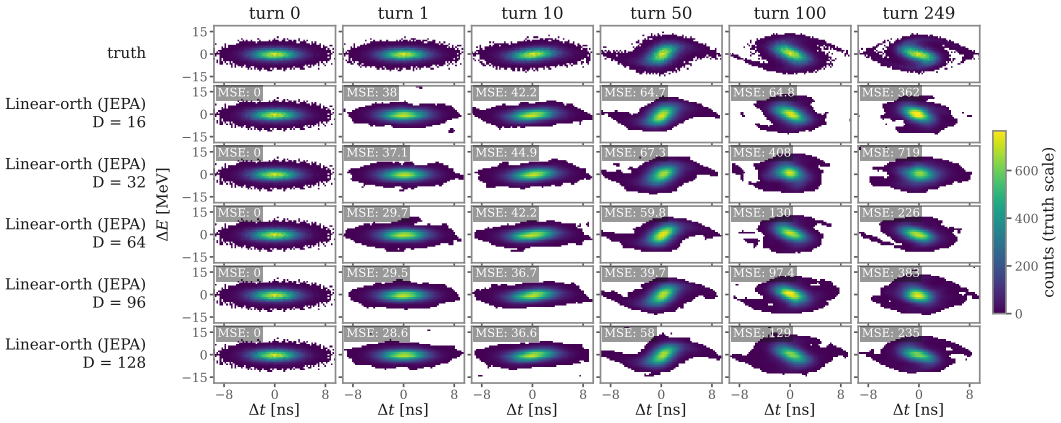

Figure 8: Autoregressive rollout of Linear-orth (JEPA) at each latent width D on the held-out simulation of Figure 3, against ground truth (top row). Columns are absolute turns; *MSE:* is each frame's error over the whole grid.

C.2 Latent Size

We retrain the arm at $D \in \{16, 32, 96, 128\}$. *If* 64 *dimensions were a bottleneck, error would keep falling past* $D = 64$*; if the width is already sufficient, larger latents would match it.*

Error falls with width up to $D = 64$ and changes little after it (Table 5). The mean rollout MSE ratio is 0.27 at $D = 16$, 0.16 at $D = 64$ and 0.14 at $D = 128$. Past $D = 64$ the gain is small: $D = 96$ and $D = 128$ lower the mean ratio by 8% and 13% and the turn-249 ratio from 0.69 to 0.62 and 0.63, and the two are within 6% of each other on every error column. The rollouts in Figure 8 show the same picture. $D = 64$ sits at the knee of the curve; doubling the latent buys at most 13% on the mean and nothing at the longest horizon, so we keep it as the default.

D Poisson Noise

The Poisson family corrupts the starting frame as

$$\tilde{o}_0 = f^{-1}\,\mathrm{Poisson}(f\,o_0)\,, \quad f = \ell^{-2}. \tag{8}$$

It redraws each bin's count with the shot noise of a simulation that tracked f times as many macro-particles, at $\ell \in \{0, 0.5, 1, 1.5, 2, 3, 4\}$. Level $\ell = 1$ is an independent redraw at the 10^5 macro-particles of the data, and $\ell = 4$ corresponds to 6,250. Empty bins stay empty.

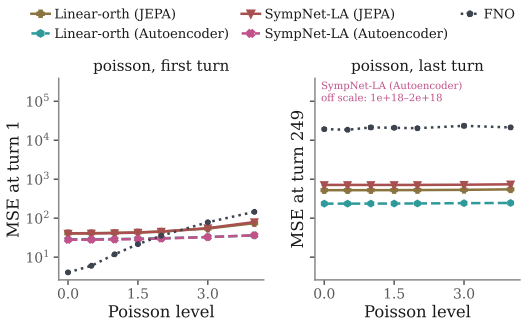

Figure 9: Robustness of predictive arms to Poisson noise in the starting frame, showing MSE at the first (left) and last (right) turn against noise level ℓ.

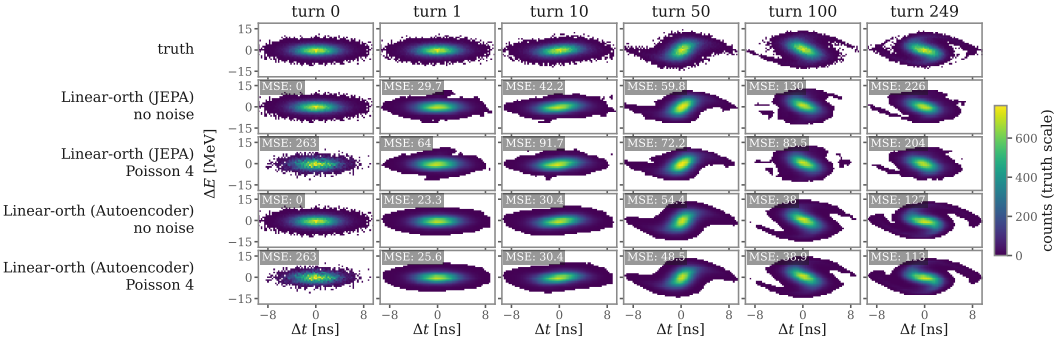

Figure 10: Phase-space rollout for Linear-orth under the JEPA and Autoencoder objectives, $\ell = 0$ vs. $\ell = 4$ Poisson noise on the starting frame, against ground truth (top row).

Poisson noise reverses the first-turn result (Figure 9, left panel). At $\ell = 4$, i.e. one sixteenth of the simulated macro-particles, the Autoencoder arms' turn-1 error rises by 31% and 26% to 36.5 and 36.1. The JEPA arms' error rises by 97% and 81% to 79.4 and 74.4, and FNO's rises $36\times$ to 145. At turn 249 neither JEPA arm nor Linear-orth (Autoencoder) moves by more than 5% across the whole range (right panel). The difference is not one of noise strength. At $\ell = 4$ and at $\sigma = 0.2$ copying the noisy frame forward gives nearly the same turn-1 error (292 and 284), yet the Autoencoder arms reach 36.5 and 36.1 under the first and 4,930 and 9,200 under the second. We attribute the difference to where the noise goes. Poisson resampling perturbs only occupied bins and leaves the background exactly empty, as it is in every training frame, whereas additive Gaussian noise also fills the background. The Autoencoder encoders tolerate noise on the bunch and fail on noise in the background, and the JEPA encoders tolerate both at the cost of a larger response to noise on the bunch. Figure 10 shows one simulation at $\ell = 4$: the noisy start roughly doubles the JEPA turn-1 error and barely changes the Autoencoder's, and both rollouts end close to their noise-free counterparts.