SyLVER: SYMBOLIC EQUATION RECOVERY FROM NEURAL POPULATION RECORDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding neural computation requires population dynamics models that are predictive and interpretable. Sequential variational autoencoders (SVAEs) recover low-dimensional latent trajectories from high-dimensional recordings but their learned dynamics are challenging to interpret. Sparse regression methods such as SINDy yield interpretable closed-form equations but require the state to be directly observed. This introduces a central challenge when applying these methods when the underlying state is unobserved: a latent state must first be inferred by a separate model, and equations are then fit to those estimates which were inferred without any knowledge of the equation. To address this challenge we introduce Symbolic Latent Variable Equation Recovery (SyLVER), a sequential VAE whose transition drift is a sparse dictionary of candidate terms that provide a closed-form dynamics equation. Because inference is structured, the drift also drives the approximate posterior through the predict step of the filter, so the dynamics and the inference network share one latent coordinate system and are trained jointly using block optimization. The recovered equation is thus constrained by inference itself rather than regressed onto post-hoc state estimates. We evaluate on the Lorenz system, van der Pol oscillator, and the MC Maze reaching dataset, and demonstrate comparable latent inference, forecasting, and coefficient recovery against two established methods, XFADS and SINDy Autoencoder.

1 INTRODUCTION

A central aim of computational neuroscience is to recover the principles by which neural populations compute. Large-scale recording deliver thousands of simultaneously monitored neurons across tasks, animals and brain areas (Jun et al., 2017; Steinmetz et al., 2021), and modeling these noisy, high-dimensional data is how hypotheses about the underlying computation are made testable. Latent variable models have been widely used for distilling population activity into low-dimensional trajectories that predict held-out neurons and behavior (Yu et al., 2009; Macke et al., 2011; Pandarinath et al., 2018; Dowling et al., 2024). The trajectories they produce, however, are governed by black-box dynamics. They predict well, but without a compact expression for the dynamics there is no natural way to compare what was learned across conditions, to design perturbations, or to ask what the model has left unexplained. This paper asks whether the dynamics of a latent variable model can instead be recovered as a sparse system of differential equations, directly from spike counts.

Empirical findings consistently show that although recordings are high-dimensional, population activity during a task typically lies near a low-dimensional manifold, and the trajectory on that manifold is well described by a dynamical system (Churchland et al., 2012; Cunningham & Yu, 2014; Gallego et al., 2017). These low-dimensional abstract descriptions are the latent variables. The computation-through-dynamics view makes them the object of interest, associating fixed points, limit cycles and flow fields with the operations the circuit performs (Sussillo & Barak, 2013; Mante et al., 2013; Vyas et al., 2020). Recovering these low-dimensional latent trajectories from spikes is a statistical inversion problem: the latent state, the dynamics that impact it, the process noise that perturbs it, and the model that links it to observations all have to be inferred jointly.

Sequential variational inference in a state-space model is an established solution to this problem (Macke et al., 2011; Archer et al., 2015; Krishnan et al., 2017; Girin et al., 2021), and with a Pois-

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.

son observation model and an amortized structured posterior, state-of-the-art models achieve high predictive accuracy (Pandarinath et al., 2018; Dowling et al., 2024). Temporal structure is a strong inductive bias that regularizes inference across time and an expressive nonlinear transition model makes forecasting possible. However, the dynamics learned by these models is opaque: A recurrent network or a multilayer drift can represent a flow field but does not expose it, so understanding what was learned require post-hoc analysis.

Alternatively, when a system admits a closed-form equation, that equation is interpretable term by term, generalizes to systems in the same dynamical family, is robust to noise, and can be updated. Sparse equation discovery methods like **SINDy** (Brunton et al., 2016) recover such equations from fully observed trajectories, but assume the state is measured, uncorrupted by noise, and in the right coordinate system.

Combining these two perspective cannot be achieved by running them in sequence: An equation fitted post hoc to inferred latents inherits whatever coordinates and errors the inference established. However, dynamics learned jointly during state inference shapes the posterior through the learned dictionary, yielding a stronger prior and residual that can be compared against alternative terms, so that a hypothesis about the dynamics can be incrementally refined.

To overcome the shortcomings of sequential variational inference and sparse equation discovery while taking advantage of their strengths, we present **SyLVER**, a sequential variational model whose transition density is a sparse dictionary inferred jointly with an amortized structured posterior. The drift sits inside the filtering recursion, so the equation being discovered is part of the posterior construction rather than a regularizer. Performing learning of dynamics and state inference jointly faces the bidirectional challenge of updating dictionary coefficients that span scales set by powers of the latent and that receive their reconstruction gradient through the filter, and inferring a posterior that is shaped by the latent-dependent dynamics. We address these with an observation-model anchor for the latent origin, a curvature-preconditioned block update for the coefficients, and masked-bin training that forces the drift to engage in state prediction. On synthetic systems SyLVER recovers the governing equations while matching the latent-recovery and forecasting accuracy of dedicated latent variable models. Finally, we test SyLVER on primate motor cortex recordings where we can infer latents from which reach velocity is decodable and recover a sparse dynamical equation for the reaching task, which to our knowledge has not previously been shown on this electrophysiological data.

2 Background

2.1 Sequential VAEs

For a partially observed system with observations $y_{1:T}$, two problems arise: inferring the latent states $z_{1:T}$ conditioned on model parameters θ and learning θ given the observations. State-space models are a standard framework for this problem. Under a Markov assumption the joint distribution of observed and unobserved variables factorizes as

$$p_\theta(y_{1:T}, z_{1:T}) = p_\theta(z_1) \prod_{t=2}^{T} p_\theta(z_t \mid z_{t-1}) \prod_{t=1}^{T} p_\theta(y_t \mid z_t), \tag{1}$$

When the transition and likelihood are linear-Gaussian, the posterior is Gaussian and computed exactly by Kalman filtering and smoothing (Särkkä, 2013). Under nonlinearity or non-conjugacy the posterior is intractable and is approximated by sampling (Doucet et al., 2000) or variational inference (Jordan et al., 1999). Amortized inference replaces per-datapoint optimization with a learned map from observations to posterior parameters (Gershman & Goodman, 2014). The resulting approximate posterior, $q_\phi(z_{1:T} \mid y_{1:T})$, is a tractable distribution over the latent states whose parameters are output by a recognition network with weights ϕ, standing in for the intractable true posterior $p_\theta(z_{1:T} \mid y_{1:T})$. The variational autoencoder (Kingma & Welling, 2014; Rezende et al., 2014) trains this map jointly with θ by maximizing the evidence lower bound (ELBO),

$$\log p_\theta(y_{1:T}) \geq \mathbb{E}_{q_\phi(z_{1:T} \mid y_{1:T})}[\log p_\theta(y_{1:T}, z_{1:T}) - \log q_\phi(z_{1:T} \mid y_{1:T})] = \mathcal{L}(\theta, \phi; y_{1:T}), \tag{2}$$

with gradients through the reparameterization trick. In the original VAE the index runs over independent data points and q_ϕ factorizes accordingly. Sequential data requires committing to a transition model and to a factorization of q_ϕ (Girin et al., 2021).

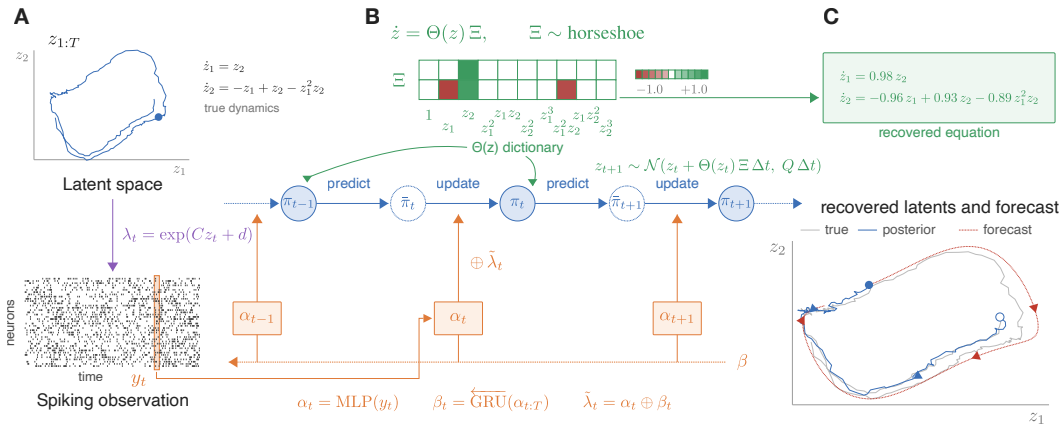

Figure 1: Overview of SyLVER on a van der Pol example. **(A)** Generative model: a low-dimensional latent trajectory $z_{1:T}$ evolves under the true dynamics and drives spike counts through a Poisson readout $\lambda_t = \exp(Cz_t + b)$. **(B)** Inference: a per-bin network $\alpha_t = \mathrm{MLP}(y_t)$ and a reverse-time GRU β_t over the sequence produce a Gaussian potential $\tilde{\lambda}_t = \alpha_t \oplus \beta_t$ in natural parameters. The posterior marginals π_t are built by a filtering recursion that alternates a predict step through the transition $z_{t+1} \sim \mathcal{N}(z_t + \Theta(z_t)\Xi\,\Delta t,\ Q)$ and an update step that adds $\tilde{\lambda}_t$. The drift is a sparse combination of dictionary terms $\Theta(z)$ with coefficients Ξ under a horseshoe prior, so the equation being learned is part of the posterior construction. **(C)** Joint training of the coefficients and the recognition network yields the recovered equation together with the inferred latents and a forecast from the learned dynamics.

2.2 Structured posterior for nonlinear dynamics

Structured variational autoencoders (Johnson et al., 2016; Zhao & Linderman, 2023) take the alternative route of keeping the prior exact and replacing only the likelihood with a learned surrogate. A recognition network emits per-timestep potentials $\psi_t(z_t)$, chosen conjugate to the latent prior (Gaussian here, even when $p_\theta(y_t \mid z_t)$ is Poisson), and the posterior is defined as

$$q(z_{1:T}) = \frac{p_\theta(z_{1:T}) \prod_t \psi_t(z_t)}{Z_\psi}, \qquad Z_\psi = \int p_\theta(z_{1:T}) \prod_t \psi_t(z_t)\, dz_{1:T}. \tag{3}$$

When this posterior is substituted into equation 2, the prior and transition terms cancel and

$$\mathcal{L}(\theta, \psi; y_{1:T}) = \sum_{t=1}^{T} \mathbb{E}_{q(z_t)}[\log p_\theta(y_t \mid z_t) - \log \psi_t(z_t)] + \log Z_\psi. \tag{4}$$

For linear-Gaussian dynamics the marginals $q(z_t)$ and Z_ψ follow from Kalman smoothing in closed form. With a nonlinear transition, the transition remains cheap to evaluate but the one-step predictive density $\bar{\pi}_t(z_t) = \int p_\theta(z_t \mid z_{t-1})\, q(z_{t-1})\, dz_{t-1}$ is no longer Gaussian. XFADS (Dowling et al., 2024) restricts this recursion to the Gaussian family by two projections per step, each the exact Kullback-Leibler (KL) minimizer within the family: 1) predict $\bar{\pi}_t = \arg\min_\pi \mathrm{KL}\big(\int p_\theta(z_t \mid z_{t-1}) q(z_{t-1})\, dz_{t-1} \,\big\|\, \pi\big)$ by moment matching, and 2) update $\pi_t \propto \bar{\pi}_t \psi_t$ by natural-parameter addition. Under this construction equation 4 collapses to

$$\mathcal{L}(\theta, \phi; y_{1:T}) = \sum_{t=1}^{T} \mathbb{E}_{\pi_t}[\log p_\theta(y_t \mid z_t)] - \sum_{t=1}^{T} \mathrm{KL}(\pi_t \,\|\, \bar{\pi}_t)\,, \tag{5}$$

where the surrogate potentials no longer appear explicitly; the recognition network is scored only through the posterior it induces, and the dynamics enter the objective through $\bar{\pi}_t$, which is where a learned transition becomes essential for inference.

2.3 Equation learning with SINDy

Discovering governing equations from data is a well-trodden problem in system identification, spanning symbolic regression via genetic programming (Bongard & Lipson, 2007; Schmidt & Lip-

son, 2009), neural and physics-guided symbolic search (Udrescu & Tegmark, 2020), and sparse regression over a dictionary of candidate functions (Brunton et al., 2016). All seek a parsimonious set of equations that explain observations. We build on the sparse identification of nonlinear dynamics (SINDy) framework (Brunton et al., 2016). Given a state $x \in \mathbb{R}^d$ and a library $\Theta(x) = [\theta_1(x), \ldots, \theta_p(x)]$ of candidate functions, SINDy posits that the dynamics are a sparse linear combination of library terms and fits the coefficients by regression on measured time derivatives,

$$\dot{x} = \Theta(x)\,\Xi, \qquad \Xi \in \mathbb{R}^{p\times d}, \tag{6}$$

$$\hat{\Xi} = \arg\min_{\Xi} \left\|\dot{X} - \Theta(X)\,\Xi\right\|_F^2 + \lambda\|\Xi\|_0, \tag{7}$$

where rows of X and $\dot{X}$ stack the state and its derivative over time. The ℓ_0 objective is approximated by sequentially thresholded least squares, which alternates least-squares fits with hard thresholding of small coefficients (Brunton et al., 2016); ℓ_1 and relaxed-regularization variants exist (Champion et al., 2020). Because equation 7 regresses on $\dot{X}$, the method requires derivatives estimated from data and is correspondingly sensitive to measurement noise, motivating weak-form (Messenger & Bortz, 2021), ensemble (Fasel et al., 2022), and Bayesian (Hirsh et al., 2022) extensions.

SINDy autoencoders (Champion et al., 2019) relax the assumption that the measured variables are the coordinates in which the dynamics are sparse. They jointly learn an encoder, decoder, and Ξ so that the latent z admits sparse dynamics, with $\dot{z}$ obtained from $\dot{x}$ by the chain rule; SINDy-SHRED (Gao et al., 2025) instead takes the hidden state of a GRU over sensor windows as z_t and embeds SINDy as the forward-Euler recurrence $z_{t+1} = z_t + \Theta(z_t)\Xi\,\Delta t$, removing the need for measured derivatives. In both, the latent is a deterministic function of the observations and the dynamics are noise-free.

The equation learning problem have also been solved probabilistically. HyperSINDy (Jacobs et al., 2023) places a variational encoder over observed states and derivatives and a hypernetwork that maps latent samples to SINDy coefficients, with sparsity from a trainable binary mask, yielding a random ODE whose coefficients are driven by white noise; the state itself is fully observed. Closest to our setting, Course & Nair (2023b) perform approximate Bayesian state estimation with a Markov Gaussian process posterior while learning the drift as a sparse dictionary expansion under a horseshoe prior from noisy observations of the state through a known observation map. Their companion work (Course & Nair, 2023a) amortizes the same posterior with an encoder and a learned decoder, but with a neural-network drift and no equation recovery. To our knowledge there is no existing method combining amortized latent inference through a learned observation model with sparse symbolic dynamics, which requires identifiability in latent-coordinate system.

3 METHODS

3.1 GENERATIVE MODEL

Let $z_t \in \mathbb{R}^d$ be the latent state and $y_t \in \mathbb{R}^n$ the observation at time bin t, with bin width Δt. The latent dynamics are the Euler–Maruyama discretisation of an SDE with drift f_Ξ and per-step transition variance Q:

$$z_1 \sim \mathcal{N}(m_0,\, P_0)\,, \tag{8}$$

$$z_{t+1} \mid z_t \sim \mathcal{N}(z_t + f_\Xi(z_t)\,\Delta t,\; Q)\,, \tag{9}$$

$$y_t \mid z_t \sim \begin{cases} \mathrm{Poisson}(\exp(Cz_t + b)) & \text{spike counts,} \\ \mathcal{N}(g(z_t),\, R) & \text{continuous observations,} \end{cases} \tag{10}$$

with P_0, Q, R diagonal and m_0, P_0 learned. The spike readout is the log-linear Poisson model (Paninski, 2004; Macke et al., 2011); g is a multilayer readout as in Dowling et al. (2024).

Following SINDy (Brunton et al., 2016), the drift is a sparse linear combination of fixed dictionary terms, $f_\Xi(z) = \Theta(z)\,\Xi$ with $\Theta(z) \in \mathbb{R}^{1\times K}$ the monomials in z up to degree p and $\Xi \in \mathbb{R}^{K\times d}$. Each coefficient carries a horseshoe prior (Carvalho et al., 2010) with a per-coefficient local scale and one global scale,

$$\xi_{kj} \mid \lambda_{kj}, \tau \sim \mathcal{N}\big(0,\, (\lambda_{kj}\tau/r_k)^2\big)\,, \qquad \lambda_{kj} \sim \mathcal{C}^+(0, a_{\mathrm{loc}}), \qquad \tau \sim \mathcal{C}^+(0, a_{\mathrm{glob}}), \tag{11}$$

where r_k standardises the k-th dictionary column. The heavy-tailed local scales let individual coefficients escape the global shrinkage, concentrating mass at zero while leaving large coefficients essentially unpenalised. Full details of the dictionary, readouts and prior are in Appendix A.1.

3.2 Latent inference and posterior structure

The posterior is constructed by recursive filtering rather than parameterizing posterior parameters directly using the variational approximation outlined in Dowling et al. (2024). Each marginal is approximated by $\pi_t = \mathcal{N}(m_t, P_t)$, with diagonal covariance. It is a deterministic function of π_{t-1}, the dynamics f_Ξ, and a data potential ψ_t. This construction allows us to learn Ξ using the gradients used to train the sequential VAE.

Data potential. ψ_t is a diagonal Gaussian in natural parameters, produced by a per-bin network shared across t and a reverse-time recurrent network over its outputs. Throughout, a diagonal Gaussian $\mathcal{N}(m, P)$ is written interchangeably in its natural parameters $(h, J) = (P^{-1}m,\ P^{-1})$, the precision-scaled mean and the precision, so that $\mathcal{N}(m, P) \propto \exp\big(h^\top z - \frac{1}{2} z^\top \mathrm{diag}(J)\, z\big)$; ψ_t is such a Gaussian with natural parameters $(\tilde{h}_t, \tilde{J}_t)$, i.e. a pseudo-likelihood with mean $\tilde{J}_t^{-1}\tilde{h}_t$ and variance $\tilde{J}_t^{-1}$.

The local potential is defined as $(h_t^\alpha,\ \ell_t^\alpha) = \mathrm{MLP}_\phi(y_t)$, and the backward potential are defined as $\left(h_t^\beta,\ \ell_t^\beta\right) = \mathrm{GRU}_\phi^{\leftarrow}(\{h_s^\alpha, \ell_s^\alpha\}_{s \geq t})$. This allows us to pass information about future time steps to compute the approximate smoothing marginal by combining local and future information $\tilde{h}_t = h_t^\alpha + h_t^\beta, \tilde{J}_t = \zeta(\ell_t^\alpha) + \zeta(\ell_t^\beta)$. Here, ζ is the softplus function, giving $\psi_t(z) \propto \exp\big(\tilde{h}_t^\top z - \frac{1}{2} z^\top \mathrm{diag}(\tilde{J}_t)\, z\big)$. The potential at t thus depends on $y_{t:T}$ and on no earlier bin; past observations reach π_t only through the predictive $\bar{\pi}_t$, matching the factorisation $\prod_t p_\theta(z_t \mid z_{t-1}, y_{t:T})$ of the exact posterior.

Predict and update. With $\mu(z) = z + f_\Xi(z)\,\Delta t$, the one-step predictive is the moment match of $\int \mathcal{N}(z_t; \mu(z_{t-1}), Q)\, \pi_{t-1}(z_{t-1})\, \mathrm{d}z_{t-1}$. Which gives us the prediction mean and precision: $\bar{m}_t = \mathbb{E}_{\pi_{t-1}}[\mu(z_{t-1})]$ and $\bar{P}_t = \mathrm{Var}_{\pi_{t-1}}[\mu(z_{t-1})] + Q$. Expectation can estimated from S reparameterised draws of π_{t-1} with Q added analytically, and the update is closed form provided conjugacy. The updates are computed using their natural parameters: $J_t = \bar{P}_t^{-1} + \tilde{J}_t,\ h_t = \bar{P}_t^{-1}\bar{m}_t + \tilde{h}_t,\ P_t = J_t^{-1},\ m_t = P_t h_t$. We use $(\bar{m}_1, \bar{P}_1) = (m_0, P_0)$ so the prior of Eq. equation 8 serves as the predictive at $t = 1$. Each step is the exact KL minimiser within the diagonal-Gaussian family. Writing the update as a precision-weighted average,

$$m_t = \frac{\bar{J}_t}{\bar{J}_t + \tilde{J}_t}\,\bar{m}_t + \frac{\tilde{J}_t}{\bar{J}_t + \tilde{J}_t}\,\frac{\tilde{h}_t}{\tilde{J}_t} \tag{12}$$

where $\tilde{h}_t/\tilde{J}_t$ is the mean of the potential, so m_t is the precision-weighted average of the dynamics prediction and the pseudo-observation; $\bar{J}_t/(\bar{J}_t + \tilde{J}_t)$ defines the share of posterior precision contributed by the dynamics; it rises as Q shrinks/gets more confident.

3.3 Objective function

The evidence lower bound is Eq. 5 with the marginals and predictives of Section 3.2,

$$\mathcal{L}(\theta, \phi) \;=\; \sum_{t=1}^{T} \mathbb{E}_{\pi_t}[\log p_\theta(y_t \mid z_t)] \;-\; \sum_{t=1}^{T} \mathrm{KL}(\pi_t \,\|\, \bar{\pi}_t)\,. \tag{13}$$

The reconstruction term is estimated by drawing $z_t \sim \pi_t$ per bin. Because π_t and $\bar{\pi}_t$ are both diagonal Gaussians, the dynamics term is available in closed form,

$$\mathrm{KL}(\pi_t \,\|\, \bar{\pi}_t) \;=\; \frac{1}{2} \sum_{j=1}^{d} \left[\frac{P_{t,j}}{\bar{P}_{t,j}} + \frac{(m_{t,j} - \bar{m}_{t,j})^2}{\bar{P}_{t,j}} - 1 - \log \frac{P_{t,j}}{\bar{P}_{t,j}} \right], \tag{14}$$

and with $(\bar{m}_1, \bar{P}_1) = (m_0, P_0)$ the $t = 1$ term is the divergence from the initial-state prior. Eq. equation 14 is a divergence between marginals rather than a sampled log-density ratio: the

posterior is defined by the recursion and has no product form over consecutive states, so $\log p_\theta(z_t \mid z_{t-1}) - \log \pi_t(z_t)$ at independently drawn z_{t-1}, z_t does not estimate the bound. The closed form also removes the estimator variance from the term through which Ξ receives its gradient.

3.4 Parameter learning and equation discovery

Parameters are split into an inference block $\phi = \{$recognition network, reverse recurrence, readout, $m_0, P_0, Q\}$ and a coefficient block $\{\Xi, \lambda, \tau\}$, updated in alternation. The loss is the negative bound of Eq. equation 13 averaged over trials plus the coefficient prior divergence,

$$\mathcal{J} = -\frac{1}{N}\sum_{n=1}^{N} \mathcal{L}_n(\theta, \phi) + \frac{1}{N}\,\mathrm{KL}(q(\Xi, \lambda, \tau) \,\|\, p(\Xi, \lambda, \tau))\,, \tag{15}$$

where $q(\Xi_{:,j}) = \mathcal{N}(\mu_j, L_j L_j^\top)$ is a full-covariance Gaussian per output dimension, the scale posteriors are the conjugate inverse-Gamma families of the augmentation in Eq. 17, and $\mathcal{L}_n$ is evaluated at one draw of Ξ shared across the batch.

Our learning method is defined by two properties. First, the dictionary coefficients live on scales set by the latent frame and its powers, with no normalisation available for them, so their update must be invariant to reparameterisation of Ξ and move along a large range of scale that first-order methods do not allow. Second, the coefficient learning signal must come through the full bound, where the reconstruction term reaches Ξ through the filter of Section 3.2. **Inference block:** One Adam step on $\mathcal{J}$ with respect to ϕ, with Ξ and the scales held fixed.

Coefficient block: With the inference block frozen, the filter is run once to obtain $\pi_{1:T}$ and $\bar{\pi}_{1:T}$, and $g_j = \nabla_{\Xi_{:,j}} \mathcal{J}$ is taken by automatic differentiation through Eq. equation 15. For each output dimension j the update is a Newton step preconditioned by a diagonal curvature estimate $\hat{H}_j$ with Levenberg–Marquardt damping (Levenberg, 1944; Marquardt, 1963),

$$\Xi_{:,j} \leftarrow \Xi_{:,j} - \frac{N}{1+\lambda}\,\hat{H}_j^{-1} g_j. \tag{16}$$

$\hat{H}_j$ has two parts. The dynamics-KL and prior contributions are analytic: with the posterior frozen the predictive mean is linear in Ξ, so the dynamics term is the weighted least-squares curvature $\Delta t^2\,\bar{\Theta}^\top W_j \bar{\Theta}$ with $W_j = 1/\bar{P}_{\cdot,j}$. The reconstruction term reaches Ξ only through the filter, we approximate its diagonal by an exponential moving average of squared gradients, the empirical-Fisher diagonal used by adaptive first-order methods (Tieleman & Hinton, 2012; Kingma & Ba, 2015; Martens, 2020), whose relationship to the Gauss–Newton curvature is discussed by Kunstner et al. (2019) to circumvent full hessian computation.

Masked-bin training: During the coefficient block the combined potential $\tilde{h}_{t:t+k}$ is blanked on randomly placed contiguous spans. $\tilde{h}_t \leftarrow \kappa_t\,\tilde{h}_t,\ \tilde{J}_t \leftarrow \kappa_t\,\tilde{J}_t,\ \kappa_t \in \{0, 1\}$. At $\kappa_t = 0$ $\pi_t = \bar{\pi}_t$: the dynamics term of Eq. equation 14 vanishes at that bin, the state there is the drift's own prediction, and the reconstruction term still runs over all T bins. This ensures if drift does not predict the model pays reconstruction cost in proportion to gap length which enforces learning dynamics by prediction.

Equation retrieval: We don't apply a threshold over the coefficients during training. The horseshoe prior promotes coefficient sparsity, and the reported equation is obtained post hoc by greedy forward selection over dictionary terms based on the objective. Horseshoe prior automatically imposes sparsity and we only do pruning at the end as explained in the appendix.

4 Results

We compare the performance of our model with XFADS (Dowling et al., 2024) for dynamics inference and prediction while comparing against SINDy AE (Champion et al., 2019) for equation recovery performance. These comparisons are to establish baselines that SyLVER is capable of posterior inference, prediction, and equation learning of known systems. We show that SyLVER is competitive compared with these benchmarks on neural data inference and equation recovery. For our comparisons, we use van der Pol oscillator, Lorenz system, and pendulum as the synthetic datasets. To demonstrate SyVLER's potential for yielding interpretable models of neural data, we

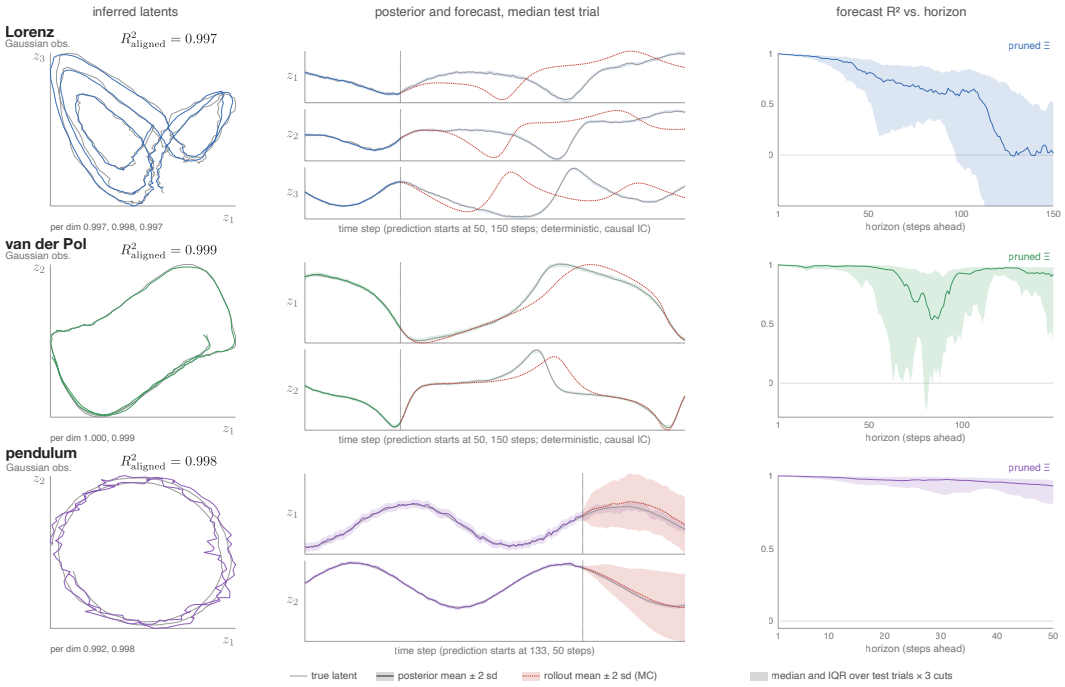

Figure 2: Latent inference and forecasting on the synthetic systems. Rows: Lorenz, van der Pol and pendulum (Gaussian observations). **Left** posterior mean trajectories of a test trial (colored) overlaid on the true latents (grey) after affine alignment, with aligned R^2 over all test trials and per-dimension values below. **Middle** the median test trial, showing the true latent, the posterior mean with ± 2 sd, and, after the dashed line, a rollout of the learned equation from the filtered state at the cut (mean ± 2 sd over Monte Carlo draws; van der Pol rollout is deterministic from a causal initial condition). Prediction starts at bin 200, 50 and 133 respectively and runs for 50, 150 and 50 bins. **Right** forecast R^2 against the true latents as a function of horizon, median and interquartile range over test trials at three cut points, using the pruned equation.

applied it to the MC Maze neural dataset (Pei et al., 2021; Churchland et al., 2010) which allows us to compare to other SOTA methods for dynamics inference and prediction under the Neural Latent Benchmark (Pei et al., 2021).

4.1 Equation recovery on synthetic data

We are able to infer the posterior and predict the van der Pol and pendulum systems nearly perfectly as can SINDY-AE and XFADS (Table 1). SyLVER is able to recover all the true terms on both datasets with spurious terms shown in red in Table 2. SINDy-AE was not able to threshold some of the smaller terms in the dictionary. This is often because of high co-linearity which is prominent in pendulum dynamics at smaller angles. For these tests we used Gaussian emission instead of Poisson. That setting works for all the models tested because it makes the problem easier than Poisson emission function but we will show that SyLVER works in that regime as well. Lorenz system is particularly challenging for SyLVER along with our run of SINDy-AE. Both the models were unable to properly learn the equation coefficients.

4.2 Equation discovery on neural data

On MC-Maze the equation that carries the forecast is dense. The learned coefficient matrix has weight across most of the 28 quadratic terms in every dimension, and the post-hoc prune to nine terms, which leaves four of six latent dimensions without any dynamics at all, drops the forecast to near zero (Figure 3D). This is the outcome the synthetic Lorenz run predicts: with an unanchored frame the sparse description of the system lives in coordinates the model did not choose, and in the

Method	van der Pol			Lorenz			Pendulum		
	R^2_{p}	R^2_{f}	$\mathrm{MSE}_{\mathrm{f}}$	R^2_{p}	R^2_{f}	$\mathrm{MSE}_{\mathrm{f}}$	R^2_{p}	R^2_{f}	$\mathrm{MSE}_{\mathrm{f}}$
SINDy-AE	99.9	94.7	0.053	99.6	87.1	0.094	99.9	99.5	0.012
XFADS	99.7	94.1	0.082	98.1	58.3	0.1	99.7	97.9	0.051
SyLVER (ours)	99.9	92.3	0.121	99.8	83.80	0.117	99.7	98.3	0.043

Table 1: Latent recovery and forecasting on synthetic datasets. R^2_{p}: coefficient of determination of the smoothed latents against the true latents after affine alignment; R^2_{f} and $\mathrm{MSE}_{\mathrm{f}}$: the same for k-step latent forecasts; LL: marginal log-likelihood of held-out observations (nats/bin). Higher is better except $\mathrm{MSE}_{\mathrm{f}}$.

Method	**van der Pol**	**Lorenz**	**Pendulum**
True	$\dot{x} = y$ $\dot{y} = -x + 1.5\,y - 1.5\,x^2 y$	$\dot{x} = -10\,x + 10\,y$ $\dot{y} = 28\,x - y - 10\,xz$ $\dot{z} = -2.67\,z + 10\,xy$	$\dot{\theta} = \omega$ $\dot{\omega} = -0.3\,\omega - 9.8 \sin\theta$
SINDy-AE	$\dot{x} = 1.23\,y$ $\dot{y} = -0.75\,x + 1.48\,y - 0.72\,x^2 y$ $+ 1$ term	$\dot{x} = -1.70\,x + 0.42\,y + 6$ terms $\dot{y} = -2.01\,x - 7.90\,y - 2.48\,xz$ $+ 10$ terms $\dot{z} = -1.04\,z + 8.18\,xy + 13$ terms	$\dot{\theta} = 1.40\,\omega + 5$ terms $\dot{\omega} = -0.23\,\omega - 3.42 \sin\theta$ $+ 7$ terms
SyLVER (ours)	$\dot{x} = 0.95\,y + 5$ terms $\dot{y} = -0.97\,x + 1.54\,y - 1.48\,x^2 y$	$\dot{x} = \varnothing_x\ \varnothing_y + 6$ terms $\dot{y} = -5.45\,y\ \varnothing_x\ \varnothing_{xz} + 4$ terms $\dot{z} = 0.71\,xy\ \varnothing_z + 4$ terms	$\dot{\theta} = 0.96\,\omega + 1$ term $\dot{\omega} = -0.37\,\omega - 10.03 \sin\theta$

Table 2: Recovered governing equations on the three synthetic systems; the true equations are given in the first row with the parameter values used (Lorenz in the rescaled frame $z_{\mathrm{native}}/10$ in which the data are generated). Only terms of the true equation are written out, to two decimals; $\varnothing$ marks a true term absent from the recovered equation, and $+\,n$ terms counts the recovered terms outside the true support. Van der Pol and pendulum SyLVER rows use a decoder fixed to the true readout so that the latent frame is the physical frame; on Lorenz no run has a pinned frame and coefficients are read in each model's own latent coordinates, identified with the physical ones up to a signed permutation. SINDy-AE's pendulum arm uses a four-frequency trigonometric bank.

model's own coordinates the same dynamics require many terms. The latents themselves are not in question. Reach velocity decodes from the smoothed posterior at $R^2 = 0.74$ with a small MLP, against 0.39 for a linear readout and 0.53 for a windowed ridge on the spikes, and the rollout from 100 ms after movement onset retains 0.73; hand position reconstructs at 0.93 from the posterior and 0.70 from the rollout. The gap between linear and nonlinear decoding says the six-dimensional latent encodes behaviour on a curved embedding rather than a linear one, which is what a polynomial drift in a learned frame should be expected to produce.

Instead of interpreting the support from the equation we try to probe its frame free structure. Freezing each latent dimension at the the beginning of rollout, we see that the forecast is carried by three of the six (z_5, z_4, z_1, together 75% of forecast R^2), so the reaching dynamics occupy a lower-dimensional subspace than the latent, consistent with the rotational, low-dimensional structure reported for motor cortex during reaching (Churchland et al., 2012). The linear block of the learned equation gives the local Jacobian, whose spectrum is invariant to the choice of latent frame. We have not analysed it here, but its complex eigenvalues would set the oscillation frequencies of the learned dynamics and could be compared directly with the 1–2 Hz rotations found by jPCA on the same task; its quadratic block is where any dependence of that rotation on the current state would appear. We regard these frame-invariant quantities as the natural objects to interpret on neural data, and a frame anchor during training, rather than post-hoc pruning, as the step that would turn the dense equation into a readable one.

5 Discussion

SyLVER learns, from spike counts alone, a latent trajectory and a sparse closed-form equation for its dynamics, with the equation discovery occurring inside the posterior construction rather than fitted post hoc. To our knowledge no previous method combines amortized latent inference through a learned observation model with simultaneous symbolic dynamics recovery: SVISE assumes a known observation map, XFADS parametrizes the drift as a neural network, and SINDy-AE makes the latent a deterministic function of the data. Making the two parts work in combination requires

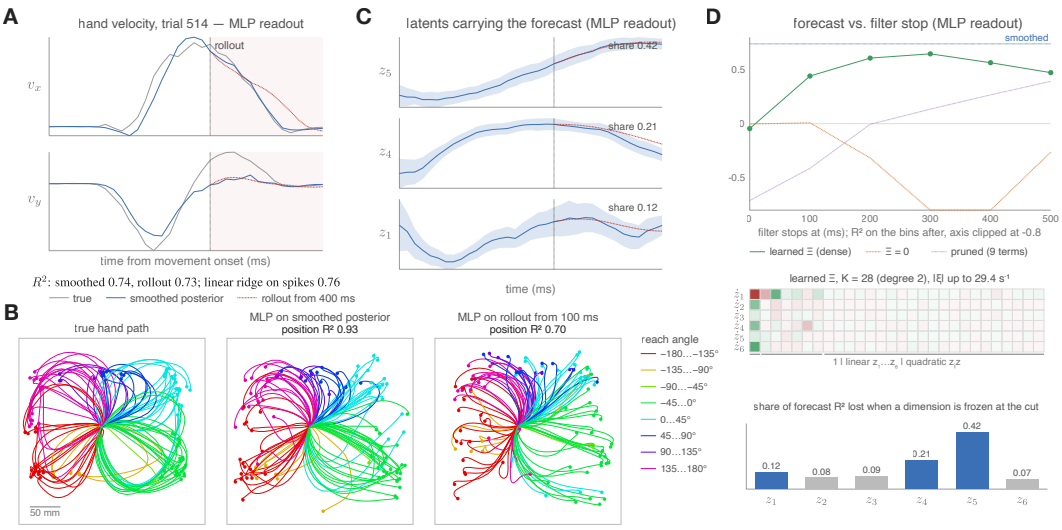

Figure 3: Reaching dynamics in primate motor cortex (MC-Maze, six latent dimensions, degree-2 dictionary). **(A)** hand velocity on a single test trial, true (grey), decoded by an MLP readout from the smoothed posterior (solid), and from a rollout of the learned equation starting 400 ms after movement onset (dotted, shaded region). Decoding R^2 over test trials: 0.74 from the smoothed posterior, 0.73 from the rollout, against 0.76 for a linear ridge readout on the spikes. **(B)** condition-averaged hand paths colored by reach angle: true, decoded from the smoothed posterior (position $R^2 = 0.93$), and decoded from a rollout starting 100 ms after onset ($R^2 = 0.70$). **(C)** the three latent dimensions carrying most of the forecast, with posterior mean $\pm$ sd and the rollout after the cut (dashed line), annotated with each dimension's share of forecast R^2. **(D)** top, decoded-velocity R^2 on the bins after the filter is stopped, as a function of the stop time, for the learned dense equation, the pruned nine-term equation, and $\Xi = 0$; the smoothed-posterior readout is shown as the reference. Middle, the learned coefficient matrix. Bottom, share of forecast R^2 lost when each latent dimension is frozen at the cut, identifying z_5, z_4 and z_1 as the dimensions the dynamics act on.

more than assembling them, and our diagnosis of why is a contribution alongside the model. On synthetic datasets the model recovers the equation and infers the posterior at a level comparable to the strongest baselines (Tables 1 and 2). On MC-Maze, velocity is poorly decoded from the latents by a linear readout but reliably decodable by an MLP, and we recover a sparse equation for the reaching task with six latent dimensions, which has not previously been shown on electrophysiological data; we expect larger latent dimension and richer dictionaries to improve it further.

Two aspects of the problem make joint learning hard. First, the latent frame is determined only up to an affine map, so any update to the frame requires a matching update to the equation. Fixing the decoder often hinders learning; a learned decoder, with the coordinate frame aligned post hoc for evaluation, works better. Second, dictionary coefficients live on scales set by powers of the latent and receive their reconstruction gradient only through the filter, and they are coupled to a posterior that they themselves shape. A curvature-preconditioned block update handles the wide range of coefficient scales, and masked-bin training breaks the coupling by forcing the drift to predict through gaps the encoder cannot fill.

The residual of a sparse fit is itself a measurement: a structured residual names what the dictionary is missing, whether an unobserved variable or a term outside the library. This suggests using the recovered equation as a diagnostic rather than a final answer, growing the latent dimension or the dictionary where the residual is structured. Two further directions follow directly from the model. Differentiable term selection, with attention over dictionary columns conditioned on the current state, would replace post-hoc pruning with a selection the posterior can learn; and because the drift is symbolic, exogenous inputs and control enter the dictionary as terms of known form, opening equation-level comparison across tasks and conditions within the same recording.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets, clean and reformat dataset, support qualitative and thematic data analysis, implement methods. We have not used generative AI tools to interpret results, help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments. and formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, , assist with translation are not applicable to this work.

Additionally, we used generative AI tools to create or modify scientific figures or images, create or edit software code, identify relevant literature, summarize or analyse existing literature, sourcing/searching for information, edit a research paper to improve readability, and brainstorming. We have reviewed all AI-assisted work. All text changes done for readability or illustration editing was double checked for any artifacts or superfluous claims. LLM-generated code was verified and tested for correctness through multiple different diagnostic tests. We cross checked literature survey with manual post-hoc curation. We also checked for potential plagiarism through a manual search. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Method details: all the essential details about parameters, hyper-parameters, learning algorithm, exact neural network descriptions, and initializations are shared in the appendix.

Data: All the data used in our paper is either synthetic with corresponding equations and details in the appendix. The neural data used is an open source dataset available on https://dandiarchive.org/dandiset/000128.

Code: The code will be made publically available through supplementary material with all the configurations, exact data files, and notebooks to replicate the results directly with the default configurations used for generating results for this work.

REFERENCES

Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.

Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007. doi: 10.1073/pnas.0609476104.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

Kathleen Champion, Bethany 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.

Kathleen Champion, Peng Zheng, Aleksandr Y. Aravkin, Steven L. Brunton, and J. Nathan Kutz. A unified sparse optimization framework to learn parsimonious physics-informed models from data. *IEEE Access*, 8:169259–169271, 2020. doi: 10.1109/ACCESS.2020.3023625.

Mark M Churchland, John P Cunningham, Matthew T Kaufman, Stephen I Ryu, and Krishna V Shenoy. Cortical preparatory activity: representation of movement or first cog in a dynamical machine? *Neuron*, 68(3):387–400, 2010.

Mark M. Churchland, John P. Cunningham, Matthew T. Kaufman, Justin D. Foster, Paul Nuyujukian, Stephen I. Ryu, and Krishna V. Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, 2012.

Kevin Course and Prasanth Nair. Amortized reparametrization: efficient and scalable variational inference for latent sdes. *Advances in Neural Information Processing Systems*, 36:78296–78318, 2023a.

Kevin Course and Prasanth B Nair. State estimation of a physical system with unknown governing equations. *Nature*, 622(7982):261–267, 2023b.

John P. Cunningham and Byron M. Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17(11):1500–1509, 2014.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

Matthew Dowling, Yuan Zhao, and Il Memming Park. exponential family dynamical systems (xfads): Large-scale nonlinear gaussian state-space modeling. *Advances in Neural Information Processing Systems*, 37:13458–13488, 2024.

Urban Fasel, J. Nathan Kutz, Bingni W. Brunton, and Steven L. Brunton. Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A*, 478(2260):20210904, 2022. doi: 10.1098/rspa.2021.0904.

Juan A. Gallego, Matthew G. Perich, Lee E. Miller, and Sara A. Solla. Neural manifolds for the control of movement. *Neuron*, 94(5):978–984, 2017.

ML Gao, JP Williams, and JN Kutz. Sparse identification of nonlinear dynamics and koopman operators with shallow recurrent decoder networks. arxiv, 2025.

Samuel J. Gershman and Noah D. Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, 2014.

Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *Foundations and Trends in Machine Learning*, 15(1–2):1–175, 2021.

Seth M. Hirsh, David A. Barajas-Solano, and J. Nathan Kutz. Sparsifying priors for Bayesian uncertainty quantification in model discovery. *Royal Society Open Science*, 9(2):211823, 2022. doi: 10.1098/rsos.211823.

Mozes Jacobs, Bingni W Brunton, Steven L Brunton, J Nathan Kutz, and Ryan V Raut. Hypersindy: Deep generative modeling of nonlinear stochastic governing equations. *arXiv preprint arXiv:2310.04832*, 2023.

Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/7d6044e95a16761171b130dcb476a43e-Paper.pdf.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

James J. Jun, Nicholas A. Steinmetz, Joshua H. Siegle, Daniel J. Denman, Marius Bauza, Brian Barbarits, Albert K. Lee, Costas A. Anastassiou, Alexandru Andrei, Çağatay Aydın, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Rahul G. Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical Fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. *Advances in neural information processing systems*, 30, 2017.

Jakob H. Macke, Lars Buesing, John P. Cunningham, Byron M. Yu, Krishna V. Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

Enes Makalic and Daniel F. Schmidt. A simple sampler for the horseshoe estimator. *IEEE Signal Processing Letters*, 23(1):179–182, 2016.

Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, 2013.

Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

Daniel A. Messenger and David M. Bortz. Weak SINDy: Galerkin-based data-driven model selection. *Multiscale Modeling & Simulation*, 19(3):1474–1497, 2021. doi: 10.1137/20M1343166.

Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D. Stavisky, Jonathan C. Kao, Eric M. Trautmann, Matthew T. Kaufman, Stephen I. Ryu, Leigh R. Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15(10):805–815, 2018.

Liam Paninski. Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15(4):243–262, 2004.

Felix Pei, Joel Ye, David Zoltowski, Anqi Wu, Raeed H Chowdhury, Hansem Sohn, Joseph E O'Doherty, Krishna V Shenoy, Matthew T Kaufman, Mark Churchland, et al. Neural latents benchmark'21: evaluating latent variable models of neural population activity. *arXiv preprint arXiv:2109.04463*, 2021.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.

Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009. doi: 10.1126/science.1165893.

Nicholas A. Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, 2021.

David Sussillo and Omri Barak. Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Computation*, 25(3):626–649, 2013.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5 – RMSProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020. doi: 10.1126/sciadv.aay2631.

Saurabh Vyas, Matthew D. Golub, David Sussillo, and Krishna V. Shenoy. Computation through neural population dynamics. *Annual Review of Neuroscience*, 43:249–275, 2020.

Matt P. Wand, John T. Ormerod, Simone A. Padoan, and Rudolf Frühwirth. Mean field variational Bayes for elaborate distributions. *Bayesian Analysis*, 6(4):847–900, 2011.

Byron M. Yu, John P. Cunningham, Gopal Santhanam, Stephen I. Ryu, Krishna V. Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, 102(1):614–635, 2009.

Yixiu Zhao and Scott Linderman. Revisiting structured variational autoencoders. In *International Conference on Machine Learning*, pp. 42046–42057. PMLR, 2023.

A Appendix

A.1 Generative model details

Transition noise convention. Q in equation 9 is the per-step transition variance, not a diffusion rate; the corresponding continuous-time diffusion is $Q/\Delta t$, and datasets simulated from $\mathrm{d}z = f(z)\,\mathrm{d}t + \sigma\,\mathrm{d}W$ correspond to $Q = \sigma^2\Delta t$. Q is a free parameter constrained to $[q_{\min}, q_{\max}]$ by projection after each optimiser step rather than by a read-time clamp, so that a saturated Q retains a nonzero gradient and can return.

Initial state. m_0 and P_0 are learned. Fixing them at $\mathcal{N}(0, I)$ makes that unit variance the only scale anchor in the latent frame and, when the true latent lives elsewhere, forces the drift to absorb the offset as a constant term.

Dictionary. $\Theta(z)$ is ordered as the constant term followed by degree-major monomials of degree 1 to p, with $\binom{d+p}{p}$ columns in total ($K = 20$ for $d = p = 3$). Trigonometric terms $\sin(\omega_r z_i), \cos(\omega_r z_i)$ over a frequency bank $\{\omega_r\}$ can be appended but are not closed under affine changes of the latent frame: $\sin(\omega(az + b))$ lies in the span only if ωa is already in the bank, so closure requires the frame change to be a signed permutation. They are not used in any experiment reported here. The coefficients entering the drift are clipped to $\pm\xi_{\max}$ as a numerical safeguard; the bound is never active in reported runs.

Column standardisation. r_k in equation 11 is the root-mean-square of the k-th dictionary column evaluated at the current posterior latents, refreshed at each coefficient update. It is a deterministic standardisation, not a third stochastic level of the prior, and makes the prior act on coefficients in units in which each column has unit scale; the physical-unit coefficient is recovered as ξ_{kj}/r_k.

Half-Cauchy scales. Each half-Cauchy scale uses the inverse-Gamma augmentation (Wand et al., 2011; Makalic & Schmidt, 2016)

$$v \mid w \sim \mathrm{IG}\left(\tfrac{1}{2}, \tfrac{1}{w}\right), \qquad w \sim \mathrm{IG}\left(\tfrac{1}{2}, \tfrac{1}{a^2}\right) \implies \sqrt{v} \sim \mathcal{C}^+(0, a), \tag{17}$$

which is exact and admits conjugate closed-form updates for the variational scale distributions (Appendix A.2). The defining property of the horseshoe, a $\mathrm{Beta}(\frac{1}{2}, \frac{1}{2})$ distribution on the shrinkage factor $\kappa = 1/(1 + \lambda^2\tau^2)$, is reproduced to a maximum quantile error of 0.0015 at $a_{\mathrm{loc}} = \tau = 1$. Under this prior the expected quadratic penalty on a coefficient with standardised second moment M behaves as $\sqrt{M} \propto |\xi|$ for $M \to 0$, an L_1-like kink at the origin, and saturates at one nat for large M, so selection is driven by the prior rather than by a threshold. A log-normal variational relaxation of the scales (Louizos et al., 2017) yields a ridge at the origin instead and does not select.

Spike readout. The rate $\exp(Cz_t + b)$ is the expected count in a bin of width Δt, not a rate in Hz. The exponential is the canonical link, giving a log-likelihood concave in (C, b). The bias is fixed at $b_i = \log$ of the mean training count of channel i; this anchors the latent origin, closing the translation gauge of the latent frame, using only the observed counts. The log-rate is clipped to $[-6, 6]$ for numerical stability.

Continuous readout. g is the sigmoid-MLP readout of with a learned diagonal observation variance $R = \mathrm{diag}(\mathrm{softplus}(\rho))$.

A.2 Inference details

Monte Carlo predict. Draws are $z_{t-1}^{(s)} = m_{t-1} + \sqrt{P_{t-1}}\,\varepsilon^{(s)}$, $\varepsilon^{(s)} \sim \mathcal{N}(0, I)$, and $\bar{P}_t$ uses the biased sample variance of $\mu(z^{(s)})$. Because Q is added analytically the recursion is defined at $S = 1$, but the propagated-uncertainty term $\mathrm{Var}_{\pi_{t-1}}[\mu]$ is then identically zero and Eq. equation **??** is not the moment match; S is a correctness parameter, not a speed parameter. Latent samples used in the reconstruction term are $z_t = m_t + \sqrt{P_t}\,\eta_t$, drawn once per bin. States and posterior means are clipped to $\pm z_{\max}$ and variances floored at 10^{-6} for numerical stability.

Reverse-time potential. The backward GRU runs right-to-left over the sequence (h^α, ℓ^α), so its hidden state at t already summarises bins $t{+}1{:}T$; a linear head on that state gives $(h_t^\beta, \ell_t^\beta)$. Both heads are zero-initialised with a negative bias on ℓ, so the potential begins near-uninformative and the dynamics prior dominates early training.

Masked bins. Gap lengths are drawn uniformly from $[g_{\min}, g_{\max}]$ with the number of gaps set to reach a target coverage ϕ (default 0.33, gaps of 5); gaps may overlap, so realised coverage is $\leq \phi$, and $t = 1$ is never masked. The mask is applied to the combined potential after the backward GRU. Masking h^α before the GRU would let h^β carry information across the gap and defeat the purpose. Under block training only the coefficient block sees the mask; the inference block does not.

A.3 Synthetic data details

All synthetic latents are simulated by Euler–Maruyama with process noise drawn once per emitted bin as $\sigma_{\mathrm{proc}}\sqrt{\Delta t}\,\varepsilon$, so the per-step transition variance is $Q_{\mathrm{true}} = \sigma_{\mathrm{proc}}^2 \Delta t$. The transient Lorenz and van der Pol datasets integrate the deterministic drift with 10 substeps per bin and still add noise once per bin at the outer Δt. Trials are independent trajectories from random initial conditions with no burn-in, so transients are kept. The readout $C \in \mathbb{R}^{n \times d}$ is drawn as $c_{\mathrm{scale}}\,\mathcal{N}(0, 1)$ entries under a fixed seed; Gaussian observations are $y_t = Cz_t + \epsilon_t$ with $\epsilon_t \sim \mathcal{N}(0, 0.1\,I)$ ($\sigma_{\mathrm{obs}} = 0.316$), and Poisson observations use $\lambda_t = \exp(Cz_t + b)$ with the log-rate clipped to $[-4, 4]$ at generation. The Gaussian variants of van der Pol and pendulum share their latent trajectories bit for bit with the Poisson variants and differ only in the readout scale and noise. Train, validation and test trials are disjoint.

System	Used in	trials (tr/te)	T	Δt	σ_{proc}	n	obs.	initial condition
Lorenz	Fig. 2	240 / 60	300	0.02	0.10	100	Poisson	$U([-40, 40]^2 \times [-20, 80])$
van der Pol	Fig. 2	400 / 50	500	0.05	0.14	100	Gaussian	$U([-3, 3]^2)$
Pendulum	Fig. 2	160 / 40	200	0.02	0.03	100	Gaussian	$\theta_0 \sim U[-2, 2], \omega_0 \sim \mathcal{N}(0, 0.3^2)$

Table 3: Synthetic dataset configurations as run. The van der Pol figure and table runs use different trial counts and initial-condition distributions. A separate validation split of 50 trials exists for the van der Pol figure run.

Lorenz attractor. $\dot{x} = \sigma(y - x)$, $\dot{y} = x(\rho - z) - y$, $\dot{z} = xy - \beta z$ with $\sigma = 10$, $\rho = 28$, $\beta = 8/3$. Each 6 s trial contains about four lobe crossings. The Poisson readout has $c_{\mathrm{scale}} = 0.012$ and $b = \log 0.316$, giving measured firing rates of 5.7 / 16.2 / 42.5 Hz at the 1st / 50th / 99th percentile (mean 17.5 Hz).

van der Pol oscillator. $\dot{x} = y$, $\dot{y} = -x + \mu(1 - x^2)\,y$ with $\mu = 1.5$. Gaussian readout with $c_{\mathrm{scale}} = 1/\sqrt{2}$.

Pendulum. $\dot{\theta} = \omega$, $\dot{\omega} = -b\,\omega - (g/\ell)\sin\theta$ with $g/\ell = 9.8$ and $b = 0.3$. The initial-condition range ($|\theta_0| \leq 2$ rad, small ω_0) keeps trials below the separatrix; the measured θ standard deviation is 0.72 rad and no trial wraps. Gaussian readout with $c_{\text{scale}} = 1/\sqrt{2}$.

A.4 Model details

Dictionary. Constant plus all monomials up to degree 3 ($K = 10$ for $d = 2$, $K = 20$ for $d = 3$). The pendulum additionally includes $\sin z_i, \cos z_i$: at unit frequency in the pinned-decoder run ($K = 14$) and over a frequency bank $\{0.5, 1, 2, 2\pi\}$ in the learned-decoder run ($K = 26$). The trigonometric block is omitted for Lorenz and van der Pol because over the latent range those systems visit, $\sin z$ and $\cos z$ are almost in the span of the cubic library and only add collinearity. MC-Maze uses degree 2 with $d = 6$ ($K = 28$).

Encoder. The per-bin network is an MLP on the encoded channels with two hidden layers of width 64, outputting $(h_t^\alpha, \ell_t^\alpha) \in \mathbb{R}^{2d}$; the precision is $\text{softplus}(\ell)$ with no clamp. The reverse-time network is a single-layer GRU of hidden size 64 run right to left over (h^α, ℓ^α), with a linear head to $(h_t^\beta, \ell_t^\beta)$. The precision heads are initialised with zero weights and a bias giving precision 0.5 per head, so the data potential starts input-independent at $\tilde{J}_t = 1.0$. On synthetic data all channels are encoded; on MC-Maze the encoder sees $n_{\text{enc}} = 137$ of 182 neurons and the remaining 45 enter only the reconstruction term.

Decoder. Poisson: linear readout $C \in \mathbb{R}^{n \times d}$ with orthonormal columns via a QR reparameterisation, initialised at a random orthonormal matrix, bias fixed at the log mean training count per channel, log-rate clipped to $[-6, 6]$. Gaussian: linear readout with learned diagonal noise $R = \text{diag}(\text{softplus}(\rho))$, initialised at $R = 0.1\,I$. Pinned-decoder arms set C to the true readout, freeze it, and learn only the bias.

Filter. Diagonal marginals; S Monte Carlo draws in the predict step; states clipped at $\pm z_{\max}$ and variances floored at 10^{-6}. m_0 and P_0 are learned from 0 and $\sigma_0^2 I$. Q is a free per-step variance initialised at q_{init} and, where bounds are set, projected to $[q_{\min}, q_{\max}]$ after each optimiser step.

Prior. Horseshoe with unit half-Cauchy scales, $a_{\text{loc}} = a_{\text{glob}} = 1$, and $\mathbb{E}[1/v]$ capped at 10^8 for numerical stability; coefficients clipped at $\pm\xi_{\max}$.

Optimisation. Inference block: one joint Adam step at learning rate 10^{-3}, gradient norm clipped at 10, weight decay 10^{-4} on the decoder only (0 on MC-Maze). Coefficient block every 2 epochs: one damped preconditioned step with the diagonal Gauss–Newton–Bartlett curvature, damping initialised at 10^2, up/down factor 3, at most 8 tries per round, per-round decay 0.9, and acceptance on the masked objective; masked-bin fraction 0.33 in gaps of exactly 5 bins. The three Poisson runs enable the held-out rollback guard every 25 rounds. The final checkpoint is reported in every case.

A.5 Full recovered equations

Table 4 gives every recovered equation behind Table 2 in full, with terms outside the true support in red. Coordinates follow the conventions of the main table: physical frame for the pinned-decoder SyLVER runs, and each model's own latent coordinates identified with the physical ones up to a signed permutation otherwise.

System	Method	Recovered equation
van der Pol	SINDy-AE	$\dot{x} = 1.23\,y$
		$\dot{y} = -0.75\,x + 1.48\,y - 0.72\,x^2 y + 0.29\,xy$
	SyLVER	$\dot{x} = 0.95\,y + 0.27\,x - 0.07\,x^3 + 0.10\,x^2 y - 0.12\,xy^2 + 0.02\,y^3$
		$\dot{y} = -0.97\,x + 1.54\,y - 1.48\,x^2 y$
Lorenz	SINDy-AE	$\dot{x} = -1.70\,x + 0.42\,y + 4.46 - 1.31\,z + 1.82\,xz - 14.20\,yz + 1.38\,xy^2 - 0.76\,xyz$
		$\dot{y} = -2.01\,x - 7.90\,y - 2.48\,xz + 0.84 + 9.23\,z + 4.56\,xy - 0.34\,yz - 1.32\,z^2 - 1.17\,xy^2 + 0.78\,xyz + 4.87\,y^3 - 4.43\,y^2 z + 1.19\,yz^2$
		$\dot{z} = -1.04\,z + 8.18\,xy + 1.37 - 0.89\,x + 1.54\,y - 1.98\,x^2 + 1.43\,xz + 0.84\,y^2 + 0.60\,z^2 - 0.39\,x^2 y + 0.20\,x^2 z - 0.40\,xyz - 0.93\,y^3 + 1.87\,y^2 z - 0.80\,yz^2$
	SyLVER	$\dot{x} = 23.97 + 0.40\,x^2 + 0.78\,xy - 1.36\,xz - 1.29\,yz + 0.03\,z^3$
		$\dot{y} = -5.45\,y - 12.01 + 9.89\,z + 0.84\,xy + 0.52\,yz - 0.04\,x^3$
		$\dot{z} = 0.71\,xy + 1.10\,y + 0.76\,x^2 - 0.50\,y^2 + 0.04\,xy^2 - 0.02\,z^3$
Pendulum	SINDy-AE	$\dot{\theta} = 1.40\,\omega - 0.37\,\theta^2\omega - 0.21\,\theta\omega^2 + 1.55\sin(0.5\theta) + 2.31\sin(0.5\omega) + 1.57\sin\theta$
		$\dot{\omega} = -0.23\,\omega - 3.42\sin\theta - 1.60\,\theta + 0.63\,\theta^2\omega + 0.35\,\theta\omega^2 - 2.71\sin(0.5\theta) - 4.64\sin(0.5\omega) + 0.20\sin\omega + 0.30\sin(2\theta)$
	SyLVER	$\dot{\theta} = 0.96\,\omega - 0.09\,\theta^3$
		$\dot{\omega} = -0.37\,\omega - 10.03\sin\theta$

Table 4: Full recovered equations for every system and method in Table 2. Terms outside the true support are shown in red. True Lorenz coefficients are in the rescaled generating frame; the constants in the SyLVER Lorenz equation are 70–89% of what the fitted affine frame map requires, i.e. largely the latent offset rather than a spurious dynamical term.