A Latent Neural ODE-VAE for Modeling Hippocampal Population Activity on Low-Dimensional Manifolds

Kathleen Higgins
University of Delaware
kathigg@udel.edu

Manuel Schottdorf
University of Delaware
maschott@udel.edu

Abstract

Neural population activity traces trajectories in a high-dimensional state space, yet accumulating evidence suggests these trajectories are confined to low-dimensional manifolds that encode both task variables and internal state. Existing manifold inference pipelines can recover geometry and explain variability, but often rely on multi-stage local models and do not impose globally smooth continuous-time dynamics. We develop a latent Neural ODE variational autoencoder (ODE-VAE) that jointly learns (i) a low-dimensional stochastic initial condition, (ii) continuous-time latent dynamics parameterized by a mixture-of-experts ODE, and (iii) a decoder back to neural activity. To better align reconstruction with temporally structured variability, we augment the variational objective with transition-consistency regularization in observation space and an LLE-inspired neighborhood reconstruction constraint in latent space. To reduce evaluation optimism on biological recordings, we split trials before fitting any normalization (and optional PCA) transforms, ensuring train-only preprocessing. On synthetic random-foraging sequences, the model achieves high reconstruction accuracy ($R^2 = 0.9789$). On the E65 hippocampal calcium dataset, a recent raw-space run (no PCA; latent dimension $D = 7$) attains Pearson correlation $r = 0.7794$ and reconstruction $R^2 = 0.6049$. Together, these results highlight both the promise and current fragility of end-to-end continuous-time manifold models for noisy biological recordings.

1 Introduction

Neural activity can be described as a point in a high-dimensional coordinate system, where each coordinate axis represents a single neuron's activity [Cunningham and Yu, 2014]. Underlying properties of the network and its inputs can confine neural trajectories to a subregion of this space, often referred to as a neural manifold [Cunningham and Yu, 2014, Gallego et al., 2017]. The neural manifold has been proposed to underlie motor movements [Gallego et al., 2017, Russo et al., 2018], head direction cells [Chaudhuri et al., 2019], and hippocampal maps of physical variables [O'Keefe and Dostrovsky, 1971, Frank et al., 2000, Wood et al., 2000, O'Keefe and Nadel, 1978]. The conceptual ideas in these studies suggest a general principle of hippocampal computation: the construction of organized maps of learned knowledge instantiated by neural manifolds [Tolman, 1948, O'Keefe and Nadel, 1978, Stachenfeld et al., 2017, Bellmund et al., 2018, Nieh et al., 2021].

Nonlinear dimensionality reduction has demonstrated that neural population activity can often be described by 4–6 latent variables, suggesting that activity is constrained to a low-dimensional neural manifold that displays a geometric representation of both physical and abstract variables [Low et al., 2018, Chaudhuri et al., 2019, Nieh et al., 2021]. Existing approaches are limited to multi-stage machine-learning pipelines, using forest-based transition models (with probabilistic principal component analysis in decision-tree leaves) to define distances between population states, which are then embedded into a low-dimensional manifold and mapped back to neural activity for reconstruction

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

[Low et al., 2018, Tipping and Bishop, 1999, Breiman, 2001, Tenenbaum et al., 2000, Yu et al., 2009]. This piecewise approach partitions state space and models dynamics locally, hence lacking explicit enforcement of globally smooth latent dynamics and can exhibit saturation of reconstruction decoding performance with low-dimensional embeddings [Low et al., 2018]. Thus, we hypothesize that generative deep learning models offer a complementary framework: neural network architectures can be trained directly on biological neural population recordings to jointly learn low-dimensional latent coordinates, their temporal evolution, and the mapping back to neural activity [Kingma and Welling, 2014, Chen et al., 2018, Rubanova et al., 2019].

In this paper, we propose a novel approach to modeling the neural manifold by constructing a Neural Ordinary Differential Equation variational autoencoder (ODE-VAE): a deep generative model that (i) encodes high-dimensional population activity into a low-dimensional latent state, (ii) models the evolution of that latent state as a continuous-time dynamical system parameterized by a neural ODE, and (iii) decodes the resulting latent trajectory back into neural activity [Kingma and Welling, 2014, Chen et al., 2018, Rubanova et al., 2019]. By training the encoder, dynamics, and decoder end-to-end under a variational objective, this approach aims to capture nonlinear manifold structure while imposing smooth temporal dynamics. Our implementation uses mixture-of-experts latent dynamics and adds two regularizers inspired by manifold inference—transition-consistency in observation space and an LLE-inspired neighborhood reconstruction constraint in latent space [Low et al., 2018, Saul and Roweis, 2003]. We evaluate this family on synthetic and hippocampal calcium datasets and analyze the sensitivity of performance to preprocessing and evaluation choices.

Contributions.

- We formalize an ODE-VAE for trialized population sequences with mixture-of-experts latent dynamics and explicit geometric regularizers.
- We implement this formulation in a versioned codebase (`v1-v5`) and highlight a current configuration with train-only preprocessing, transition consistency, and an LLE-inspired neighborhood reconstruction regularizer.
- We provide a reproducible evaluation on synthetic and hippocampal calcium datasets and identify protocol factors that strongly affect reconstruction metrics.

2 Related Work

Our approach lies at the intersection of manifold-based neuroscience and latent dynamical systems. In hippocampus, the cognitive map framework and subsequent experimental work motivate geometric organization of population codes [Tolman, 1948, O'Keefe and Dostrovsky, 1971, O'Keefe and Nadel, 1978, Eichenbaum and Cohen, 2014], including abstract and non-spatial representations [Aronov et al., 2017, Tavares et al., 2015, Constantinescu et al., 2016, Schuck and Niv, 2019, Park et al., 2020, Nieh et al., 2021]. Beyond classical place coding, hippocampal population activity reflects trajectory and sequential organization [Frank et al., 2000, Pastalkova et al., 2008, MacDonald et al., 2011, Taxidis et al., 2020], episodic variables at shared locations [Wood et al., 2000, Gill et al., 2011, McKenzie et al., 2014], and multimodal/task variables such as odor and taste [Eichenbaum et al., 1987, Herzog et al., 2019]. Manifold inference methods can recover low-dimensional structure and explain structured variability beyond measured task variables [Low et al., 2018, Chaudhuri et al., 2019, Rubin et al., 2019].

In machine learning, variational autoencoders [Kingma and Welling, 2014] and neural ODEs [Chen et al., 2018] provide a principled framework for continuous-time latent-variable modeling. Latent ODEs extend this idea to irregularly sampled sequences [Rubanova et al., 2019]. We adopt this framework but tailor the encoder, evaluation protocol, and regularization to the neuroscience setting, emphasizing trialized sequences, explicit geometric constraints, and comparisons to MIND-style evaluation pipelines [Low et al., 2018]. For calcium imaging recordings, related methodological work has emphasized motion correction and demixing/denoising [Pnevmatikakis et al., 2016, Pnevmatikakis and Giovannucci, 2017], highlighting the importance of preprocessing choices when evaluating reconstruction metrics.

3 Problem Setup and Data

We study trialized population activity sequences. Let $y_b(t_\ell) \in \mathbb{R}^N$ denote the raw activity of N simultaneously recorded units/ROIs on trial $b \in \{1, \dots, B\}$ at resampled time t_ℓ, where $\ell \in \{1, \dots, L\}$ indexes a fixed-length grid. We write $Y_b \in \mathbb{R}^{L \times N}$ for the stacked sequence.

Observation space. The implementation supports training either in raw ROI space or in a PCA-compressed space. Let $x_b(t_\ell) \in \mathbb{R}^P$ denote the activity vector used by the model after preprocessing, where $P = N$ for raw space and $P = K$ when PCA is enabled. To avoid leakage, all normalization (and optional PCA) transforms are fit on training data only and then applied to validation/test sequences. Unless otherwise stated, reported metrics (r, R^2) are computed in the evaluation space associated with each run.

Time grid. Trials are resampled to a common duration and the time vector is normalized to $[0, 1]$; we denote the resulting grid by $0 = t_1 < \cdots < t_L = 1$. The latent dimension is denoted by D.

E65 dataset. We use the Schottdorf Lab E65 dataset (`E65_data.npz`), containing calcium activity ($\Delta F/F$) from $N = 375$ ROIs over $T = 7434$ frames, along with trial IDs and timestamps. Frames are grouped by trial, the first 10 trials are dropped, and each trial is linearly interpolated to a fixed length $L = 120$ (`trial_len_s=12`, `fps=10`), with time normalized to $[0, 1]$. After filtering, 180 trials are available.

Our default "no leakage" preprocessing is trial-split-first: (i) construct trial sequences Y_b; (ii) optionally subsample to 100 sequences via greedy landmark selection for speed; (iii) split trials into train/validation (default: hold out the last 3 sequences); (iv) fit preprocessing transforms on training only (per-feature normalization across all training frames and optional PCA), and apply the same transforms to validation. During training, an optional per-trial baseline is removed by subtracting the mean of the first 5 resampled bins. Concretely, for training trials we compute per-feature moments

$$\mu = \frac{1}{B_{\mathrm{tr}} L} \sum_{b \in \mathcal{T}_{\mathrm{tr}}} \sum_{\ell=1}^{L} y_b(t_\ell), \qquad \sigma = \sqrt{\frac{1}{B_{\mathrm{tr}} L} \sum_{b \in \mathcal{T}_{\mathrm{tr}}} \sum_{\ell=1}^{L} \big(y_b(t_\ell) - \mu\big)^{\odot 2} + \varepsilon}, \tag{1}$$

and normalize all trials as $x_b(t_\ell) = (y_b(t_\ell) - \mu) \oslash \sigma$, where $\oslash$ and $\odot$ denote elementwise division and multiplication and $\varepsilon = 10^{-8}$. If baseline correction is enabled, we further set $\tilde{x}_b(t_\ell) = x_b(t_\ell) - \frac{1}{5} \sum_{j=1}^{5} x_b(t_j)$.

Synthetic benchmark. We additionally evaluate on `synthetic_rat_data.npz` (4000 frames, 300 neurons, 20 trials), which provides a controlled benchmark for recoverability of smooth low-dimensional dynamics.

4 Model: Latent Neural ODE-VAE

4.1 Stochastic encoder

For each trial, an encoder network parameterizes a diagonal Gaussian posterior on the latent initial state. Let $X_b = [x_b(t_1), \dots, x_b(t_L)] \in \mathbb{R}^{L \times P}$ denote the preprocessed trial sequence in the model's observation space. The approximate posterior is

$$q_\phi(z_{0,b} \mid X_b) = \mathcal{N}\big(\mu_b, \mathrm{diag}(\sigma_b^2)\big), \tag{2}$$

with reparameterization

$$z_{0,b} = \mu_b + \sigma_b \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{3}$$

Here $z_{0,b} \in \mathbb{R}^D$, $\mu_b \in \mathbb{R}^D$, $\sigma_b \in \mathbb{R}^D_{>0}$, and $\odot$ denotes elementwise multiplication. In our current configuration, $(\mu_b, \log \sigma_b^2)$ are produced by a Transformer sequence encoder applied to X_b, but MLP and recurrent encoders are also supported.

4.2 Continuous-time latent dynamics

Latent trajectories are generated by a neural ODE:

$$\frac{dz_b(t)}{dt} = f_\theta(z_b(t)), \qquad z_b(t_1) = z_{0,b}. \tag{4}$$

We parameterize f_θ as a mixture of experts:

$$f_\theta(z) = \sum_{e=1}^{E} \pi_e(z) f_e(z), \qquad \pi(z) = \mathrm{softmax}(g(z)), \tag{5}$$

In the reported experiments, this defines an autonomous ODE (no explicit dependence on t); time dependence can be introduced by conditioning the gate and experts on a learned embedding of t. We use $E = 4$ latent experts by default. Each expert $f_e : \mathbb{R}^D \to \mathbb{R}^D$ is an MLP and $\pi_e(z) \in [0, 1]$ are gating weights satisfying $\sum_e \pi_e(z) = 1$.

4.3 Decoder family

A decoder maps latent states back to observations:

$$\hat{x}_b(t_\ell) = g_\psi(z_b(t_\ell)). \tag{6}$$

The codebase supports MLP, neuron-aware, local-attention, and mixture-of-experts (MoE) decoders; our current configuration uses an MoE decoder with 8 decoder experts. In all cases, $g_\psi : \mathbb{R}^D \to \mathbb{R}^P$ outputs the mean of a factorized Gaussian observation model in the chosen observation space.

5 Training Objective and Regularization

We optimize a variational objective with auxiliary regularizers. Under a Gaussian observation model $p_\psi(x_b(t_\ell) \mid z_b(t_\ell)) = \mathcal{N}(g_\psi(z_b(t_\ell)), \sigma^2 I)$ with fixed σ^2, maximizing the standard ELBO ($\beta = 1$) corresponds (up to constants and a scale factor) to minimizing mean-squared reconstruction error plus a KL penalty. In our experiments we use a weighted-KL variant (β-VAE), with β_t warmed up to a final β.

The base objective combines reconstruction and KL terms:

$$\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{rec}} + \beta\, \mathcal{L}_{\text{KL}}, \tag{7}$$

where

$$\mathcal{L}_{\text{rec}} = \frac{1}{B\,L\,P} \sum_{b=1}^{B} \sum_{\ell=1}^{L} \|\hat{x}_b(t_\ell) - x_b(t_\ell)\|_2^2, \tag{8}$$

$$\mathcal{L}_{\text{KL}} = \frac{1}{B} \sum_{b} D_{\text{KL}}\big(q_\phi(z_{0,b} \mid X_b) \,\|\, \mathcal{N}(0, I)\big). \tag{9}$$

Equivalently, the (negative) β-VAE objective per trial is

$$\mathcal{L}_{\beta\text{-VAE}} = -\mathbb{E}_{q_\phi(z_{0,b}|X_b)}\Big[\sum_{\ell=1}^{L} \log p_\psi(x_b(t_\ell) \mid z_b(t_\ell))\Big] + \beta\, D_{\text{KL}}\big(q_\phi(z_{0,b} \mid X_b) \,\|\, p(z_{0,b})\big), \tag{10}$$

which reduces to the negative ELBO when $\beta = 1$. We use prior $p(z_{0,b}) = \mathcal{N}(0, I)$. In practice, the code uses a single Monte Carlo sample of $z_{0,b}$ per trial and minibatch.

Smoothness regularization.

$$\mathcal{L}_{\text{smooth}} = \frac{1}{B(L-1)D} \sum_{b,\ell} \left\| \frac{z_b(t_{\ell+1}) - z_b(t_\ell)}{t_{\ell+1} - t_\ell} \right\|_2^2. \tag{11}$$

Transition-aware regularization.

$$\mathcal{L}_{\text{trans}} = \frac{1}{B(L-1)P} \sum_{b,\ell} \big\| \big(\hat{x}_b(t_{\ell+1}) - \hat{x}_b(t_\ell)\big) - \big(x_b(t_{\ell+1}) - x_b(t_\ell)\big) \big\|_2^2. \tag{12}$$

This term is linearly warmed up for the first 30 epochs.

LLE-inspired neighborhood reconstruction regularization. For flattened latent points $\{z_i\}_{i=1}^{M_{\text{LLE}}} \subset \mathbb{R}^D$, with k-NN set $\mathcal{N}_k(i)$, we add a neighborhood reconstruction penalty inspired by locally linear embedding (LLE) [Saul and Roweis, 2003]:

$$\mathcal{L}_{\text{LLE}} = \frac{1}{M_{\text{LLE}} D} \sum_{i=1}^{M_{\text{LLE}}} \left\| z_i - \sum_{j \in \mathcal{N}_k(i)} w_{ij} z_j \right\|_2^2, \quad w_{ij} = \frac{\exp\left(-\frac{\|z_i - z_j\|_2}{\tau}\right)}{\sum_{j' \in \mathcal{N}_k(i)} \exp\left(-\frac{\|z_i - z_{j'}\|_2}{\tau}\right)}. \tag{13}$$

Unlike classical LLE, we do not solve for per-point constrained least-squares weights; instead we use kernel-normalized weights, yielding a simple differentiable local-consistency regularizer. Default parameters: $k = 8$, $M_{\text{LLE}} \leq 256$, $\tau = 0.1$.

Total loss.

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \beta_t \mathcal{L}_{\text{KL}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}} + \lambda_{\text{trans},t} \mathcal{L}_{\text{trans}} + \lambda_{\text{LLE}} \mathcal{L}_{\text{LLE}}. \tag{14}$$

The KL coefficient β_t is warmed up over 30 epochs to a final value $\beta = 0.02$.

6 Versioned Model Development and Failure Modes

The repository contains a sequence of incrementally modified training scripts that reflect both model and pipeline iteration: the current implementation lives in `v5_neural_vae.py` and earlier variants are archived under `neuroscience/src/archived_ode_models/`. These versions should not be interpreted as a clean ablation study: they differ in architecture, preprocessing, evaluation space, solver choices, and logging conventions. Nevertheless, documenting this evolution is useful for understanding observed fragilities and for motivating the transition and neighborhood reconstruction regularizers, which are directly inspired by the geometry- and transition-aware components of the MIND pipeline [Low et al., 2018, Saul and Roweis, 2003].

`v1`: baseline latent Neural ODE-VAE. `v1` implements the minimal continuous-time VAE setup [Kingma and Welling, 2014, Chen et al., 2018]: an MLP encoder of $x(t_1)$, a single latent vector field f_θ, an MLP decoder, and an ℓ_2 smoothness penalty on finite-difference latent velocities. It also introduces global PCA preprocessing and greedy landmark selection, mirroring common manifold inference practice [Low et al., 2018]. *Vulnerabilities:* (i) PCA is fit on the full recording before the train/val split, which can leak test-set structure into the representation; (ii) landmark selection is performed on flattened time points and mapped back to trials via a modulo operation, which can duplicate trials and bias the subsample away from true trial-level coverage; (iii) the default ordered holdout (holding out the last few trials) is sensitive to nonstationarities or ordering effects. *Motivation for `v2`:* reduce protocol-induced optimism by splitting first and fitting normalization/PCA only on training data, and explore more expressive dynamics.

`v2`: switching/gated latent dynamics with train-only preprocessing. `v2` adds a learned gating network over multiple candidate latent vector fields (a switching/mixture-style dynamics), increasing expressivity beyond a single global f_θ. Crucially, `v2` builds raw trial sequences first, then performs the train/val split, and fits both standardization and PCA on training data only; evaluation reconstructs back to raw ROI space via inverse PCA before scoring, which is closer to MIND-style reporting [Low et al., 2018]. *Vulnerabilities:* (i) time normalization to $[0, 1]$ is disabled in the script, which changes the effective scale seen by the ODE solver and can make optimization more sensitive; (ii) gating dynamics introduce additional nonconvexity and can collapse to a single expert without careful tuning. *Motivation for `v3`:* incorporate mixture-of-experts dynamics and richer decoders to better capture heterogeneous neural tuning and trial-to-trial variability.

`v3`: mixture-of-experts latent ODE and decoder variants. `v3` introduces a mixture-of-experts latent vector field (soft gating over multiple f_e) and a family of decoders (neuron-aware, local-attention, and MoE decoders) intended to better model neuron-specific heterogeneity. It also adds an optional per-trial baseline correction (subtracting early-bin means) to reduce drift/offset burden on the latent state. *Vulnerabilities:* (i) the preprocessing/evaluation pipeline reverts to full-session PCA and PCA-space scoring, making results harder to compare to raw-space metrics and potentially optimistic; (ii) the flattened-time landmark subsampling and ordered-holdout issues from `v1` persist; (iii) added model capacity increases overfitting risk when validation is extremely small. *Motivation* *for v4:* refine the decoder locality bias and improve hardware compatibility (notably Apple MPS) while keeping the MoE latent dynamics.

`v4`**: MoE latent dynamics with locality-biased decoders (MPS-safe).** `v4` largely preserves `v3`'s MoE latent dynamics and decoder choices, and emphasizes MPS-safe ODE integration (fixed-step fallbacks) to reduce device-specific solver failures during experimentation. *Vulnerabilities:* decoder "locality" is primarily architectural (attention-like) rather than enforced by an explicit geometric objective, and the pipeline-level issues (PCA leakage, trial subsampling bias, small/ordered holdout) remain. *Motivation for v5:* add explicit constraints that directly regularize temporal transitions and local manifold geometry, closer in spirit to MIND's use of transition structure and neighborhood geometry [Low et al., 2018, Saul and Roweis, 2003].

`v5`**: transition consistency + neighborhood reconstruction regularization.** The current model retains MoE latent dynamics and MoE decoding while adding two explicit regularizers: (i) a transition-consistency loss in observation space that matches $\Delta\hat{x}(t)$ to $\Delta x(t)$ and (ii) an LLE-inspired neighborhood reconstruction penalty that encourages each latent point to be reconstructible from its k-NN neighborhood [Saul and Roweis, 2003]. Both are motivated by the observation that reconstruction alone can ignore fine-grained temporal and local geometric structure, which MIND leverages via transition-aware distances and local mappings [Low et al., 2018]. *Vulnerabilities:* (i) the default training script still subsamples training data using flattened-time landmark selection and uses a small ordered validation set, amplifying sensitivity to preprocessing and random seed; (ii) the codebase supports multiple evaluation spaces (PCA vs raw), so reported R^2 values are not directly comparable unless the evaluation definition is matched.

7 Experimental Protocol

7.1 Configurations

Main settings (from `neuroscience/configs/v5_base.txt` and saved run configs): latent dimension $D = 7$, batch size 8, 150 epochs, Adam optimizer (learning rate 0.001, weight decay 10^{-5}), KL coefficient $\beta = 0.02$ with a 30-epoch warmup, $\lambda_{\text{smooth}} = 10^{-3}$, $\lambda_{\text{trans}} = 5 \times 10^{-3}$ with a 30-epoch warmup, $\lambda_{\text{LLE}} = 5 \times 10^{-3}$, landmark count 100, and baseline correction enabled. To avoid preprocessing leakage, we enable train-only preprocessing (trial split before fitting normalization and optional PCA transforms).

Implementation details. The current configuration uses a lightweight Transformer sequence encoder (hidden width 256, 1 layer, 2 heads) with pooling on the first token to parameterize $q_\phi(z_0 \mid X)$. The latent vector field uses $E = 4$ experts with hidden width 128 and a learned gating network; derivatives are layer-normalized for stability. For reconstruction, we use a mixture-of-experts decoder with 8 decoder experts and hidden width 256. Latent dynamics are integrated with Dormand–Prince (`dopri5`) using tolerances `rtol`=10^{-2} and `atol`=10^{-3}. Gradients are clipped to max norm 1.0.

7.2 Metrics

We report both Pearson correlation coefficient and coefficient of determination on validation sequences. For a validation set, we compute

$$r = \operatorname{corr}\Big(\operatorname{vec}(X_{\text{val}}), \operatorname{vec}(\hat{X}_{\text{val}})\Big), \tag{15}$$

where $\operatorname{vec}(\cdot)$ denotes vectorization over trial, time, and feature dimensions. We additionally compute coefficient of determination,

$$R^2 = 1 - \frac{\sum_{b,\ell} \|x_b(t_\ell) - \hat{x}_b(t_\ell)\|_2^2}{\sum_{b,\ell} \|x_b(t_\ell) - \bar{x}\|_2^2}, \tag{16}$$

where $\bar{x} = \frac{1}{B\,L} \sum_{b,\ell} x_b(t_\ell)$ denotes the mean activity vector across all validation entries in the evaluation space. Metrics can be computed either in raw ROI space or in PCA space. When PCA is enabled, raw-space scoring is obtained by applying inverse PCA and de-normalization to $\hat{x}$ before computing r and R^2, which makes comparisons to MIND-style reconstruction metrics more direct [Low et al., 2018].

Table 1: E65 reconstruction metrics under different training/evaluation spaces. "Dim" denotes embedding dimension d for MIND and latent dimension D for ODE-VAE. PCA-space metrics are not directly comparable to raw ROI metrics.

Method	Train	Eval	Dim	r	R^2
MIND [Low et al., 2018]	PCA	raw	7	0.7237	0.5236
ODE-VAE (PCA $\rightarrow$ raw)	PCA	raw	7	0.6658	0.4010
ODE-VAE (raw $\rightarrow$ raw)	raw	raw	7	0.7794	0.6049
ODE-VAE (PCA $\rightarrow$ PCA)	PCA	PCA	7	0.8627	0.7241

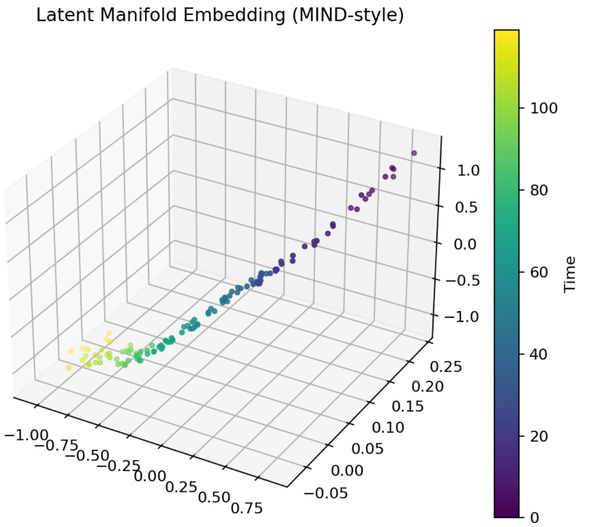

Figure 1: Latent manifold embedding produced by the ODE-VAE analysis pipeline.

8 Results

8.1 E65 hippocampal data: effect of preprocessing and evaluation space

Runs in this repository differ not only in model capacity but also in *where* reconstruction is optimized and *where* metrics are computed (raw ROI space vs. PCA space). Table 1 reports the latest E65 metrics for three closely matched ODE-VAE runs (all share the same hyperparameters and no-leakage preprocessing) that differ only by whether PCA is applied (`use_pca`) and whether metrics are computed in raw vs. PCA space (`eval_metrics_pca`). Training in PCA space but scoring in raw ROI space (`2026-02-22_155529_e7f3cc6`) yields $R^2 = 0.4010$, whereas training and scoring in raw ROI space (no PCA; `2026-02-22_160802_b03d3ed`) improves to $R^2 = 0.6049$. Scoring in PCA space (`2026-02-22_155909_e7f3cc6`) produces higher values ($R^2 = 0.7241$), but these are not directly comparable to raw-space metrics because PCA compresses and reweights variance. For context, Table 1 also includes a MIND baseline [Low et al., 2018]; note that MIND uses a random 90/10 split while our default ODE-VAE evaluation holds out the last 3 trials.

8.2 Manifold interpretability

The codebase saves latent manifold projections (MDS) and reconstruction diagnostics for each run. Figure 1 shows an example latent trajectory embedding from the trained model artifacts.

9 Discussion

The model captures the intended inductive bias: low-dimensional continuous latent trajectories with explicit geometric regularization. On synthetic data, this bias is highly effective. On real E65 recordings, however, results are sensitive to implementation and evaluation choices.

Three factors emerge from the saved run artifacts:

1. **Metric-space mismatch.** PCA-space training can look favorable while strict raw-space R^2 may degrade. This is particularly salient when comparing to MIND-style evaluations, which reconstruct back to neuron space (via inverse PCA) before scoring [Low et al., 2018].
2. **Data-efficiency tradeoff.** Landmark subsampling (100 selected sequences from 180 usable trials) accelerates training but may reduce generalization. In MIND, landmarks primarily support graph construction and embedding; the learned mapping is then applied to all eligible time points [Low et al., 2018].
3. **Optimization stability.** Strong regularization with small validation sets (3 trials) and stiff latent dynamics can produce unstable or negative final R^2, despite early high points.

These observations suggest that future gains likely require protocol-level changes in addition to architectural changes: larger and randomized holdout splits, early stopping on a stable cross-validated objective, trial-level (not frame-level) landmark selection, and direct raw-space reconstruction losses.

10 Future Work

A central motivation of this project is to connect end-to-end continuous-time latent dynamical modeling with the multi-stage manifold inference pipeline used in MIND [Low et al., 2018]. Our current codebase already adopts several MIND-inspired components (global PCA preprocessing, greedy landmark selection for visualization, and MDS-based manifold plots), but the modeling philosophy differs: MIND estimates a graph of transition structure via a PPCA regression forest and learns explicit local mappings between ambient activity and manifold coordinates [Low et al., 2018, Tipping and Bishop, 1999, Breiman, 2001], whereas the ODE-VAE learns a single global generative model (encoder + latent dynamics + decoder) by optimizing a reconstruction objective [Kingma and Welling, 2014, Chen et al., 2018, Rubanova et al., 2019]. Below we outline concrete directions to tighten this connection and improve robustness on calcium recordings.

10.1 Match MIND-style evaluation protocols and metrics

Many apparent discrepancies across saved E65 runs are consistent with evaluation-definition mismatch. In the MIND Matlab cross-validation script, trials are split randomly (e.g., 90/10), reconstruction is scored in the original neuron space after mapping back through inverse PCA, and performance is visualized both as an overall score and as per-trial dots [Low et al., 2018]. Aligning our training and reporting with this protocol would make comparisons substantially more interpretable. Concretely, we plan to (i) report both Pearson correlation on vectorized activity blocks,

$$r = \mathrm{corr}\Big(\mathrm{vec}(Y_{\mathrm{test}}), \mathrm{vec}(\hat{Y}_{\mathrm{test}})\Big)\,, \tag{17}$$

and variance-explained R^2 under repeated random trial splits, and (ii) include held-out neuron evaluation where latents are inferred from a subset of neurons and used to predict excluded neurons, mirroring the "cell prediction" analyses in MIND [Low et al., 2018]. This will also require revisiting the current practice of validating on the final 3 trials, which can conflate generalization with drift.

10.2 Use landmarks for geometry, not for shrinking the training set

In MIND, landmarks are an efficiency device for graph construction and embedding; the learned mapping is then applied to all eligible time points [Low et al., 2018]. In contrast, the default configuration further subsamples the dataset down to 100 landmarked sequences (from 180 trials), which likely increases estimator variance and can bias which trials are emphasized during training. A straightforward next step is to train the ODE-VAE on all trials/time points and reserve landmark selection for: (i) visualization, (ii) neighbor graph construction for local regularizers, and (iii) lightweight geometric diagnostics (e.g., random-walk distance embeddings). This change should directly improve stability without changing the model class.

10.3 Hybrid decoders: combine global reconstruction with MIND-like local mappings

The MIND pipeline learns mappings between ambient PCA space and manifold coordinates using locally weighted methods (e.g., LLE regression) [Saul and Roweis, 2003, Low et al., 2018]. This provides a natural mechanism to capture sharp, local irregularities that global regressors may smooth out. Our current decoders are global function approximators (MLP/MoE), which can yield good coarse reconstructions but may miss neuron-specific transients. An appealing hybrid is a global decoder plus a local residual term defined over nearby latent states,

$$\hat{x}(t) = g_{\psi}(z(t)) + \sum_{j \in \mathcal{N}_k(z(t))} \alpha_j(z(t))\, r_j, \tag{18}$$

where $\mathcal{N}_k(\cdot)$ are neighbors in latent space (or in a MIND-style random-walk metric), $r_j \in \mathbb{R}^K$ are learned prototype residuals, and α_j are normalized weights (e.g., softmax over distances). This would preserve the interpretability and global smoothness of the ODE while injecting the kind of local adaptivity that MIND's mapping stage provides.

10.4 Optimize and score in raw neuron space (with PCA as an internal linear layer)

Several E65 runs in this repository train and score in different spaces (PCA vs raw ROI), making R^2 values hard to compare. MIND keeps PCA primarily as a compression step but reconstructs back to the original activity space before computing reconstruction scores [Low et al., 2018]. A direct analogue for the ODE-VAE is to keep a fixed (or lightly fine-tuned) PCA projection for computational efficiency, but decode back to raw ROI space and compute the main reconstruction loss on $y_b(t) \in \mathbb{R}^N$. One implementation is to parameterize a raw-space decoder as $\hat{y}(t) = W_{\text{PCA}}^{\top}\hat{x}(t) + \mu$, using the PCA loading matrix W_{PCA} and mean μ from preprocessing, and to define $\mathcal{L}_{\text{rec}}$ in raw space. This would more closely match the scientific question—reconstructing neural activity—and reduce the chance that good PCA-space fits hide biologically relevant errors.

10.5 Make latent dynamics probabilistic to better match MIND transition structure

MIND estimates transition structure via a probabilistic model of next-step activity (a PPCA regression forest) and then derives a random-walk geometry from transition probabilities [Low et al., 2018, Tipping and Bishop, 1999, Breiman, 2001]. Our latent ODE is deterministic given $z_{0,b}$, which can be brittle when real data exhibit unmodeled inputs, nonstationarities, or observation noise. A natural extension is to introduce process noise (Neural SDEs) or discrete-time stochastic residuals, $z(t_{\ell+1}) = z(t_\ell) + \int_{t_\ell}^{t_{\ell+1}} f_\theta(z(t))\, dt + \eta_\ell$, which can absorb variability not explained by the initial condition while retaining smooth latent structure. This direction also creates a clearer conceptual bridge between ODE-based dynamics and MIND's transition-probability graph.

10.6 Geometry-aware objectives beyond neighborhood reconstruction

Our current neighborhood reconstruction penalty encourages local consistency in the learned latent point cloud, but it does not directly use transition structure. The MIND code constructs local distances from transition probabilities (e.g., $d_{ij} \propto \sqrt{-\log p_{ij}}$) and then computes geodesic distances on the resulting graph before embedding [Low et al., 2018]. This is conceptually related to geodesic-distance embeddings in nonlinear dimensionality reduction [Tenenbaum et al., 2000]. A promising direction is to import this idea as a regularizer: estimate a transition graph among landmarked latent points, compute a random-walk geodesic distance matrix, and penalize distortions between these distances and Euclidean distances in the latent embedding. Such a constraint could encourage the latent representation to respect the sequential structure that MIND leverages, while still permitting an end-to-end generative model.

11 Limitations and Reproducibility

This study is bounded by the available run artifacts and inherits version-specific logging differences. In particular, some run files report "best" and "final" R^2 under different conditions, and not all checkpoints include identical metadata fields. We therefore report values exactly as saved in each artifact path. The implementation also exhibits training fragility (including occasional NaN divergence), which should be addressed before drawing definitive biological conclusions.

12 Conclusion

We presented a mathematically grounded latent Neural ODE-VAE framework for neural manifold modeling and analyzed a sequence of model variants (`v1-v5`). The method can recover smooth low-dimensional dynamics and high synthetic reconstruction quality, but real-data performance remains sensitive to preprocessing and evaluation protocol. This work provides a formal foundation and concrete directions for improving robustness of ODE-VAE manifold modeling for neuroscience.

References

Dmitriy Aronov, Rachel Nevers, and David W. Tank. Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature*, 543(7647):719–722, 2017.

Jacob L. S. Bellmund, Peter Gärdenfors, Edvard I. Moser, and Christian F. Doeller. Navigating cognition: Spatial codes for human thinking. *Science*, 362(6415):eaat6766, 2018.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Rishidev Chaudhuri, Burak Gerçek, Bikash Pandey, Adrien Peyrache, and Ila Fiete. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nature Neuroscience*, 22:1512–1520, 2019.

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

Alexandra O. Constantinescu, Jill X. O'Reilly, and Timothy E. J. Behrens. Organizing conceptual knowledge in humans with a grid-like code. *Science*, 352(6292):1464–1468, 2016.

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

Howard Eichenbaum and Neal J. Cohen. Can we reconcile the declarative memory and spatial navigation views on hippocampal function? *Neuron*, 83(4):764–770, 2014.

Howard Eichenbaum, Menachem Kuperstein, Andrew Fagan, and Janet Nagode. Cue-sampling and goal-approach correlates of hippocampal unit activity in rats performing an odor-discrimination task. *Journal of Neuroscience*, 7:716–732, 1987.

Loren M. Frank, Emery N. Brown, and Matthew Wilson. Trajectory encoding in the hippocampus and entorhinal cortex. *Neuron*, 27:169–178, 2000.

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.

P. R. Gill, Sheri J. Y. Mizumori, and David M. Smith. Hippocampal episode fields develop with learning. *Hippocampus*, 21:1240–1249, 2011.

Lauren E. Herzog et al. Interaction of taste and place coding in the hippocampus. *Journal of Neuroscience*, 39:3057–3069, 2019.

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

Ryan J. Low, Sean Lewallen, Dmitriy Aronov, Rachel Nevers, and David W. Tank. Probing variability in a cognitive map using manifold inference from neural dynamics. *bioRxiv*, 2018. doi: 10.1101/418939.

Chris J. MacDonald, Kyle Q. Lepage, Uri T. Eden, and Howard Eichenbaum. Hippocampal "time cells" bridge the gap in memory for discontiguous events. *Neuron*, 71:737–749, 2011.

Sam McKenzie et al. Hippocampal representation of related and opposing memories develop within distinct, hierarchically organized neural schemas. *Neuron*, 83:202–215, 2014.

Edward H. Nieh et al. Geometry of abstract learned knowledge in the hippocampus. *Nature*, 595 (7865):80–84, 2021.

John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175, 1971.

John O'Keefe and Lynn Nadel. *The Hippocampus as a Cognitive Map*. Clarendon Press, 1978.

Sang Ah Park, David S. Miller, Hamed Nili, Charan Ranganath, and Erie D. Boorman. Map making: Constructing, combining, and inferring on abstract cognitive maps. *Neuron*, 107(6):1226–1238.e8, 2020.

Eva Pastalkova, Vladimir Itskov, A. Amarasingham, and Gyorgy Buzsaki. Internally generated cell assembly sequences in the rat hippocampus. *Science*, 321:1322–1327, 2008.

Eftychios A. Pnevmatikakis and Andrea Giovannucci. Normcorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. *Journal of Neuroscience Methods*, 291:83–94, 2017.

Eftychios A. Pnevmatikakis et al. Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron*, 89:285–299, 2016.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Amir Rubin et al. Revealing neural correlates of behavior without behavioral measurements. *Nature Communications*, 10:1–14, 2019.

Abigail A. Russo et al. Motor cortex embeds muscle-like commands in an untangled population response. *Neuron*, 97(4):953–966.e8, 2018.

Lawrence K. Saul and Sam T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

Nicolas W. Schuck and Yael Niv. Sequential replay of nonspatial task states in the human hippocampus. *Science*, 364(6447):eaaw5181, 2019.

Kimberly L. Stachenfeld, Matthew M. Botvinick, and Samuel J. Gershman. The hippocampus as a predictive map. *Nature Neuroscience*, 20(11):1643–1653, 2017.

Rita M. Tavares et al. A map for social navigation in the human brain. *Neuron*, 87:231–243, 2015.

Jiannis Taxidis et al. Differential emergence and stability of sensory and temporal representations in context-specific hippocampal sequences. *Neuron*, 108:984–998.e9, 2020.

Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B*, 61(3):611–622, 1999.

Edward C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4):189–208, 1948.

Eric R. Wood, Paul A. Dudchenko, Rebekka J. Robitsek, and Howard Eichenbaum. Hippocampal neurons encode information about different types of memory episodes occurring in the same location. *Neuron*, 27:623–633, 2000.

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.