Langevin Flows for Modeling Neural Latent Dynamics

Yue Song[1], T. Anderson Keller[2], Yisong Yue[1], Pietro Perona[1], and Max Welling[3]

[1] Caltech, Vision Lab, Pasadena, CA
[2] The Kempner Institute for the Study of Natural and Artificial Intelligence, Harvard University, Cambridge, MA
[3] University of Amsterdam, Institute for Informatics, Amsterdam, The Netherlands

Abstract

Neural populations exhibit latent dynamical structures that drive time-evolving spiking activities, motivating the search for models that capture both intrinsic network dynamics and external unobserved influences. In this work, we introduce LangevinFlow, a sequential Variational Auto-Encoder where the time evolution of latent variables is governed by the underdamped Langevin equation. Our approach incorporates physical priors — such as inertia, damping, a learned potential function, and stochastic forces — to represent both autonomous and non-autonomous processes in neural systems. Crucially, the potential function is parameterized as a network of locally coupled oscillators, biasing the model toward oscillatory and flow-like behaviors observed in biological neural populations. Our model features a recurrent encoder, a one-layer Transformer decoder, and Langevin dynamics in the latent space. Empirically, our method outperforms state-of-the-art baselines on synthetic neural populations generated by a Lorenz attractor, closely matching ground-truth firing rates. On the Neural Latents Benchmark (NLB), the model achieves superior held-out neuron likelihoods (bits per spike) and forward prediction accuracy across four challenging datasets. It also matches or surpasses alternative methods in decoding behavioral metrics such as hand velocity. Overall, this work introduces a flexible, physics-inspired, high-performing framework for modeling complex neural population dynamics and their unobserved influences.

Keywords: neural population dynamics; variational auto-encoders; latent variable models

Introduction

Neural populations have been demonstrated to possess an underlying dynamical structure which drives the time evolution of population spiking activities (Shenoy et al., 2013; Vyas et al., 2020). Uncovering these underlying latent 'factors' governing neural variability has become a goal of increasing interest in the neuroscience community. Such factors have been shown to be predictive of held-out neurons, future neural dynamics, and even behavior (Gallego et al., 2017). Recent works in this field have emphasized the importance of being able to model both internal deterministic dynamics, and potentially unobserved external influences (such as input from sensory areas, or stochastic influences from other unmeasured brain regions). In established frameworks such as AutoLFADS (Pandarinath et al., 2018a), such influences have been captured by separately inferred control variables which modulate the dynamics of the inferred latent variables. Separate work has further modeled neural activity and particularly decision-making, through the use of learned potential functions that shape attractor-like population dynamics (Genkin et al., 2023). Their work revealed a single decision variable embedded in a higher-dimensional population code, where heterogeneous neuronal firing could be explained by diverse tuning to the same latent process. Notably, the notion of an attractor mechanism aligns with the concept of a potential landscape, wherein neural trajectories evolve within an energy basin that facilitates stable or quasi-stable states. In parallel, recent developments in Transformer architectures (Ye & Pandarinath, 2021; Ye et al., 2024) offer a promising avenue for neural data modeling by capturing long-range dependencies and global context across entire sequences – complementing traditional methods that focus on local temporal interactions.

Drawing from physics, the Langevin equation is a stochastic differential equation which describes a system driven by both deterministic forces and stochastic environmental influences. We propose that the Langevin equation naturally integrates the key ingredients highlighted in prior studies: intrinsic (autonomous) dynamics, unobserved external or stochastic influences, and a potential function to shape attractor-like behavior. Specifically, we introduce a novel latent variable model for neural data that leverages underdamped Langevin dynamics to describe the time evolution of latent factors. This model includes terms representing inertia, damping, a potential function, and stochastic forces arising from both internal and external sources. Crucially, the potential function in our model is parameterized as a network of locally coupled oscillators, inducing a bias towards oscillatory and flow-like dynamics previously observed in neural latent activity (Churchland et al., 2012). This formulation captures the autonomous dynamics inherent to neural systems, providing a principled way to model both the stability and variability observed in neural responses. The oscillatory potential function also mirrors the emergence of cortical rhythms and traveling waves that have been linked to critical computational roles such as information integration, synchronization, and flexible sensorimotor processing (Ermentrout & Kleinfeld, 2001; Buzsaki, 2006).

We train the model as a sequential Variational Auto-Encoder (VAE) (Kingma & Welling, 2013) with a recurrent encoder and a small one-layer transformer serving as the generative map from latent variables to neural spike rates. The recurrent encoder effectively captures local temporal dependencies in the neural data, while the Transformer decoder is employed to harness global context. By attending to the entire latent sequence, the Transformer refines firing rate predictions through integrating information from all timesteps, ensuring that long-range interactions and subtle dynamical patterns are well captured. This combination allows the model to capture complex temporal patterns and spatial correlations within the neural population data. Empirically, we first show the efficacy of our LangevinFlow on synthetic neural population data generated from a Lorenz attractor system, where our method is able to predict the firing rates closer to the ground truth than existing competitive baselines. We then demonstrate state-of-the-art performance on the Neural Latents Benchmark (NLB) (Pei et al., 2021), achieving superior results in modeling held-out neuron likelihoods (co-smoothing, bits per spike) and forward prediction accuracy across all four benchmark datasets (`MC_Maze`, `MC_RTT`, `Area2_Bump`, and `DMFC_RSG`), sampled at both 5 and 20 ms. The model also performs comparably or better in decoding behavioral metrics such as hand velocity. Notably, the time evolution of latent representations reveals smooth spatiotemporal wave dynamics, which is reminiscent of traveling waves observed in cortical activity (Muller et al., 2018). This suggests that our coupled oscillator potential might capture key computational principles underlying neural information integration. Ultimately, we present this Langevin dynamics framework for neural data modeling, which incorporates inductive biases from physical principles and accounts for unobserved influences through its inherent stochastic dynamics. This general framework also allows for the flexible design of potential functions, opening up new doors for experimentation with latent dynamical systems.

Related Work

Neural population modeling has emerged as a key area in computational neuroscience, primarily driven by technological advances that now allow us to simultaneously record from hundreds or even thousands of neurons (Stevenson & Kording, 2011). Rather than focusing on individual neurons in isolation, population-level analyses seek to uncover the collective dynamics that shape brain function. These methods aim to capture moment-to-moment variability (Churchland et al., 2006; Ecker et al., 2010), shed light on network-wide interactions (Cohen & Kohn, 2011; Saxena & Cunningham, 2019), and relate neural activity to behavior in real time (Gallego et al., 2018, 2020; Dabagia et al., 2023) — all of which are central goals for both fundamental neuroscience research and applied domains such as brain-computer interfaces (Sussillo, Stavisky, et al., 2016; Karpowicz et al., 2022).

Early approaches to analyzing population neural recordings primarily focused on relatively simple statistical or latent-variable methods. Among the most widely used are linear and switching linear dynamical systems (LDS and SLDS) (Macke et al., 2011; Kao et al., 2015; Gao et al., 2016; Linderman et al., 2017), which model neural population activity via linear state transitions (or piecewise linear segments) and emissions. Gaussian process-based approaches (Yu et al., 2008; Zhao & Park, 2017; Wu et al., 2017; Duncker & Sahani, 2018) impose smoothness assumptions on latent factors and allow flexible, nonparametric modeling. However, the need for trial-averaging and the limited expressiveness of linear or Gaussian process latent variables can miss richer structures inherent in neural data, particularly during dynamic and nonlinear brain computations. To overcome these limitations, recurrent neural network (RNN)-based methods have emerged as powerful tools to capture the non-linear dynamics (Zhao & Park, 2016; Duncker et al., 2019). One seminal work in this space is Latent Factor Analysis via Dynamical Systems (LFADS) (Sussillo, Jozefowicz, et al., 2016), which utilizes RNNs to model autonomous dynamics in single trials of spiking activity. LFADS infers latent trajectories that explain observed neural variability and has demonstrated impressive gains over traditional baselines. Subsequent work such as AutoLFADS (Pandarinath et al., 2018a) refined this framework by allowing the model to separately infer putative "control" inputs, thereby accounting for unobserved external influences (*e.g.,* sensory input or cognitive factors) that modulate neural dynamics. Following the advances in machine learning, recent work has begun exploring Transformer-based architectures for neural data. Transformers process input tokens in parallel, enabling potentially faster training and inference compared to sequential RNNs. Their success in large-scale language tasks has motivated adaptations such as the Neural Data Transformer (NDT) (Ye & Pandarinath, 2021) which modifies the Transformer encoder for neural spiking data, the improved version NDT2 (Ye et al., 2024) which further improves scaling across heterogeneous contexts, and POYO (Azabou et al., 2024) which leverages both cross-attention and PerceiverIO (Jaegle et al., 2022) to construct a latent tokenization method for neural population activities.

The most relevant methods to our work are AutoLFADS (Pandarinath et al., 2018a) and NDT (Ye & Pandarinath, 2021). AutoLFADS and LFADS employ RNNs as the encoder and decoder networks, and the temporal dynamics are given by the hidden states, while NDT uses Transformers to encode the spiking data and additionally adopts masked modeling methodology to learn the context information. By contrast, our LangevinFlow employs a recurrent encoder, an oscillatory potential to enforce Langevin dynamics to the time evolution of latent variables, and a single Transformer layer to decode the entire variable sequence to firing rates.

Methodology

In this section, we first introduce the underdamped Langevin equation, then present the sequential VAE framework, followed by the derivation and analysis of how Langevin dynamics evolve in the posterior flow of latent variables. Finally, we discuss the model architecture and the training algorithm.

Underdamped Langevin Equation

We seek to build a latent variable model which integrates the desired beneficial inductive biases (intrinsic dynamics, stochastic influences, and an attractor-like potential function) in a principled manner. From the physics literature, a canonical abstract model of a system interacting with its environment is the Langevin equation:

$$\frac{\partial \boldsymbol{z}}{\partial t} = \boldsymbol{v}, \;\; m\frac{\partial \boldsymbol{v}}{\partial t} = F(\boldsymbol{z}) - m\gamma \boldsymbol{v} + \sqrt{2m\gamma k_B \tau}\boldsymbol{\eta}(t) \tag{1}$$

where $\boldsymbol{z}(t)$ denotes the (d-dimensional) state of the system at time t, $\boldsymbol{v}$ represents the associated velocity, m is a diagonal matrix of masses, F is the set of internal forces acting on the system (as a function of its state), γ is the damping (or friction) coefficient, k_B is the Boltzmann constant, τ is the temperature, and $\boldsymbol{\eta}(t)$ represents high-dimensional Gaussian white noise modeling the thermal fluctuation.

One method for defining the force field F is in terms of the gradient of a scalar potential function $F(\mathbf{z}) = -\nabla_{\mathbf{z}} U(\mathbf{z})$. This formulation allows for the description of many well-known physical systems which have intrinsic dynamics. One abstraction of neural dynamics is that of a network of locally coupled oscillators (Diamant & Bortoff, 1969; Ermentrout & Kopell, 1984), which admits a particularly simple potential function:

$$U(\boldsymbol{z}) = \boldsymbol{z}^T \frac{\boldsymbol{W}_z}{||\boldsymbol{W}_{\boldsymbol{z}}||_2}\boldsymbol{z} \tag{2}$$

where $\boldsymbol{W}_{\boldsymbol{z}} \in \mathbb{R}^{d \times d}$ is the symmetric matrix of coupling coefficients between the individual oscillators. For a locally coupled system, this matrix reduces to a convolution operator in the Toeplitz form. Driven by this coupled oscillator potential, the time evolution of the latent state vector $\boldsymbol{z}$ will have smooth spatiotemporal oscillatory dynamics (see Fig. 3).

Sequential Variational Auto-Encoder

To leverage the Langevin equation in a latent variable model of neural data, we assert that the sequence of observed spikes $\bar{\boldsymbol{x}}$ is Poisson distributed according to the firing rate $\bar{\boldsymbol{r}}$:

$$p(\bar{\boldsymbol{x}}|\bar{\boldsymbol{r}}) = \sum_{t=0}^{T} \text{Poisson}(\boldsymbol{x}_t|\boldsymbol{r}_t) \tag{3}$$

The firing rate is predicted by a decoder which takes as input the latent state variables, detailed later. For the input sequence $\bar{\boldsymbol{x}}$, latent samples $\bar{\boldsymbol{z}}$, and sample velocities $\bar{\boldsymbol{v}}$, we further assert the following factorization of their joint distribution:

$$\begin{aligned} p(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}) &= p(\boldsymbol{v}_0)p(\boldsymbol{z}_0)\prod_{t=1}^{T} p(\boldsymbol{v}_t)p(\boldsymbol{z}_t)\prod_{t=0}^{T} p(\boldsymbol{x}_t|\boldsymbol{z}_t, \boldsymbol{v}_t) \\ &= p(\boldsymbol{v}_0)p(\boldsymbol{z}_0)\prod_{t=1}^{T} p(\boldsymbol{v}_t)\delta(\boldsymbol{z}_t - f_{\boldsymbol{z}}(\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1}))\prod_{t=0}^{T} p(\boldsymbol{x}_t|\boldsymbol{z}_t, \boldsymbol{v}_t) \end{aligned} \tag{4}$$

where $\delta(\cdot)$ denotes the Dirac δ-function, and $f_{\boldsymbol{z}}$ denotes the coupled Hamiltonian update which is introduced later in Eq. (11). Since $\boldsymbol{z}_t$ and $\boldsymbol{v}_t$ are coupled, the update to $\boldsymbol{z}$ is deterministic. We thus only define $\boldsymbol{z}_0$ and use δ-functions to represent the later deterministic transformations. Here $p(\boldsymbol{z}_0)$ and $p(\boldsymbol{v}_t)$ are both standard Normal distributions, and $p(\boldsymbol{x}_t|\boldsymbol{z}_t, \boldsymbol{v}_t)$ defines the mapping from latents to observations.

We employ the framework of Variational Autoencoders (VAEs) (Kingma & Welling, 2013), extended to sequential data, to perform inference over latent variables in this generative model. The goal of learning is to optimize the parameters of the following set of approximate posterior distributions:

$$\begin{aligned} q_\theta(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}}) &= q_\theta(\boldsymbol{z}_0, \boldsymbol{v}_0|\boldsymbol{x}_0)q(\boldsymbol{z}_{1:T}, \boldsymbol{v}_{1:T}|\boldsymbol{z}_0, \boldsymbol{v}_0) \\ &= q(\boldsymbol{z}_0|\boldsymbol{x}_0)q(\boldsymbol{v}_0|\boldsymbol{x}_0)\prod_{t=1}^{T} q(\boldsymbol{z}_t, \boldsymbol{v}_t|\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1}) \\ &= q(\boldsymbol{z}_0|\boldsymbol{x}_0)q(\boldsymbol{v}_0|\boldsymbol{x}_0)\prod_{t=1}^{T} q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1})q(\boldsymbol{v}_t|\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1}) \\ &= q(\boldsymbol{z}_0|\boldsymbol{x}_0)q(\boldsymbol{v}_0|\boldsymbol{x}_0)\prod_{t=1}^{T} \delta(\boldsymbol{z}_t - f_{\boldsymbol{z}}(\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1}))q(\boldsymbol{v}_t|\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1}) \end{aligned} \tag{5}$$

where $q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1})$ and $q(\boldsymbol{v}_t|\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1})$ are the successive conditionals for updating $\boldsymbol{z}_t$ and $\boldsymbol{v}_t$ at each timestep, respectively. Since the joint update of $\boldsymbol{z}_t$ and $\boldsymbol{v}_t$ is chosen to be autonomous, we omit later $\boldsymbol{x}_t$ for simplifying above posterior. We derive the lower bound to model evidence (ELBO) as:

$$\begin{aligned} \log p(\bar{\boldsymbol{x}}) &= \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}}))}\left[\log \frac{p(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{v}})}{q(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}})}\frac{q(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}})}{p(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}})}\right] \\ &\geq \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}}))}\left[\log \frac{p(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{v}})}{q(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}})}\right] \\ &= \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}}))}\left[\log \frac{p(\bar{\boldsymbol{x}}, \bar{\boldsymbol{v}}, \boldsymbol{z}_0|\boldsymbol{z}_{1:T})}{q(\boldsymbol{z}_0, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}}, \boldsymbol{z}_{1:T})}\frac{p(\boldsymbol{z}_{1:T})}{q(\boldsymbol{z}_{1:T}|\boldsymbol{z}_0, \boldsymbol{v}_{0:T-1})}\right] \\ &= \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}}))}\left[\log \frac{p(\bar{\boldsymbol{x}}, \bar{\boldsymbol{v}}, \boldsymbol{z}_0|\boldsymbol{z}_{1:T})}{q(\boldsymbol{z}_0, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}}, \boldsymbol{z}_{1:T})}\cancel{\frac{\prod_{t=1}^{T}\delta(\boldsymbol{z}_t - f_{\boldsymbol{z}}(\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1}))}{\prod_{t=1}^{T}\delta(\boldsymbol{z}_t - f_{\boldsymbol{z}}(\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1}))}}\right] \\ &= \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}}))}\left[\log \frac{p(\bar{\boldsymbol{x}}|\bar{\boldsymbol{v}}, \bar{\boldsymbol{z}})p(\bar{\boldsymbol{v}})p(\boldsymbol{z}_0)}{q(\boldsymbol{z}_0)q(\bar{\boldsymbol{v}}|\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}})}\right] \\ &= \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}})}\left[\log p(\bar{\boldsymbol{x}}|\bar{\boldsymbol{v}}, \bar{\boldsymbol{z}})\right] + \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}, \bar{\boldsymbol{v}}|\bar{\boldsymbol{x}})}\left[\log \frac{p(\boldsymbol{z}_0)}{q(\boldsymbol{z}_0)}\frac{p(\bar{\boldsymbol{v}})}{q(\bar{\boldsymbol{v}}|\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}})}\right] \end{aligned} \tag{6}$$

Factorizing the joint distribution over timesteps, we can further re-write the above ELBO as:

$$\begin{aligned} \log p(\bar{\boldsymbol{x}}) \geq \sum_{t=0}^{T} \mathbb{E}_{q_\theta}\left[\log p(\boldsymbol{x}_t|\boldsymbol{z}_t, \boldsymbol{v}_t)\right] \\ -\mathbb{E}_{q_\theta}\left[\mathrm{D}_{\mathsf{KL}}\left[q_\theta(\boldsymbol{z}_0|\boldsymbol{x}_0)||p(\boldsymbol{z}_0)\right]\right] - \mathbb{E}_{q_\theta}\left[\mathrm{D}_{\mathsf{KL}}\left[q_\theta(\boldsymbol{v}_0|\boldsymbol{x}_0)||p(\boldsymbol{v}_0)\right]\right] \\ -\sum_{t=1}^{T} \mathbb{E}_{q_\theta}\left[\mathrm{D}_{\mathsf{KL}}\left[q_\theta(\boldsymbol{v}_t|\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1})||p(\boldsymbol{v}_t)\right]\right] \end{aligned} \tag{7}$$

where the first term is the reconstruction objective, the second and third terms define the KL divergence on the initial distribution of latent variables, and the last term regularizes the time evolution of the posterior.

Recall that we assume that the observed spikes $\bar{\boldsymbol{x}}$ are samples from a Poisson process with underlying rates $\bar{\boldsymbol{r}}$. At each time step, the firing rates $\boldsymbol{r}_t$ are predicted as a function of the latent variables $\boldsymbol{z}_t$ and $\boldsymbol{r}_t$ from the approximate posterior. The corresponding optimization objectives are defined as:

$$\begin{aligned} \mathcal{L}^{\text{Poisson}} &= -\sum_{t=0}^{T} \mathbb{E}_{q_\theta}\left[\log\Big(\text{Poisson}(\boldsymbol{x}_t|\boldsymbol{r}_t)\Big)\right] \\ \mathcal{L}^{KL} &= \mathbb{E}_{q_\theta}\left[\mathrm{D}_{\mathsf{KL}}\Big(q_\theta(\boldsymbol{z}_0|\boldsymbol{x}_0)||p(\boldsymbol{z}_0)\Big)\right] + \mathbb{E}_{q_\theta}\left[\mathrm{D}_{\mathsf{KL}}\Big(q_\theta(\boldsymbol{v}_0|\boldsymbol{x}_0)||p(\boldsymbol{v}_0)\Big)\right] \\ &\quad + \sum_{t=1}^{T} \mathbb{E}_{q_\theta}\left[\mathrm{D}_{\mathsf{KL}}\Big(q_\theta(\boldsymbol{v}_t|\boldsymbol{z}_{t-1}, \boldsymbol{v}_{t-1})||p(\boldsymbol{v}_t)\Big)\right] \end{aligned} \tag{8}$$

where $\mathcal{L}^{\text{Poisson}}$ denotes the Poisson negative log-likelihood, and $\mathcal{L}^{KL}$ represents the KL divergence regularization. In practical implementation, the decoder q also incorporates the encoder's hidden states as part of its input. For brevity, we defer the details and provide a more thorough explanation when introducing the model architecture and referring to Fig. 1.

Latent Langevin Posterior Flow

To derive the time evolution of the posterior, we can decompose the underdamped Langevin equation into two steps:

$$\begin{aligned} &\texttt{Deterministic Step:} \quad \frac{d\boldsymbol{z}}{dt} = \boldsymbol{v}, \frac{d\boldsymbol{v}}{dt} = -\frac{\nabla_{\boldsymbol{z}} U(\boldsymbol{z})}{m} \\ &\texttt{Probabilistic Step:} \quad \frac{d\boldsymbol{v}}{dt} = -\gamma\boldsymbol{v} + \sqrt{2m\gamma k_b \tau}\boldsymbol{\eta}(t) \end{aligned} \tag{9}$$

Here the deterministic step amounts to the Hamiltonian flow where the total energy is conserved, and the probabilistic step follows the stochastic Ornstein-Uhlenbeck process. Our goal is to derive the time evolution of joint posterior probability.

Hamiltonian Flow. The deterministic step of Eq. (9) actually defines the Hamiltonian of the system:

$$\mathcal{H}(\boldsymbol{v}, \boldsymbol{z}) = \underbrace{\frac{U(\boldsymbol{z}, \boldsymbol{x})}{m}}_{\text{Potential}} + \underbrace{\frac{1}{2}||\boldsymbol{v}||^2}_{\text{Kinetic}} \tag{10}$$

The total energy is conserved in the time evolution of the coupled variables. Discretizing over time leads to the joint update:

$$\begin{aligned} [\boldsymbol{z}_{t+\frac{1}{2}}, \boldsymbol{v}_{t+\frac{1}{2}}] &= f(\boldsymbol{z}_t, \boldsymbol{v}_t) = [f_z, f_v] \\ &= [\boldsymbol{z}_t + \frac{\partial \mathcal{H}}{\partial \boldsymbol{v}_t}\Delta t, \boldsymbol{v}_t - \frac{\partial \mathcal{H}}{\partial \boldsymbol{z}_t}\Delta t] \end{aligned} \tag{11}$$

where the subscript t denote the time index, Δt represents the step size in physical time, and $[f_{\boldsymbol{z}_t}, f_{\boldsymbol{v}_t}]$ denotes the above coupled transformation. The joint posterior $q(\boldsymbol{z}_{t+1/2}, \boldsymbol{v}_{t+1/2})$ obeys the normalizing-flow-like density evolution:

$$\log q(\boldsymbol{z}_{t+\frac{1}{2}}, \boldsymbol{v}_{t+\frac{1}{2}}) = \log q(\boldsymbol{z}_t, \boldsymbol{v}_t) + \log|\det(I + \mathcal{J}_{\mathcal{H}}\Delta t)|^{-1} \tag{12}$$

where $\mathcal{J}_{\mathcal{H}}$ is the Jacobian induced by the Hamiltonian. For infinitesimal steps Δt, we have:

$$\begin{aligned} \det(I + \mathcal{J}_{\mathcal{H}}\, dt) &= \det\left[I + \begin{pmatrix} \frac{\partial^2 \mathcal{H}}{\partial z_i \partial v_j} & -\frac{\partial^2 \mathcal{H}}{\partial z_i \partial z_j} \\ \frac{\partial^2 \mathcal{H}}{\partial v_i \partial z_j} & -\frac{\partial^2 \mathcal{H}}{\partial v_i \partial z_j} \end{pmatrix} \Delta t\right] \\ &\approx 1 + \operatorname{Tr}\begin{pmatrix} \frac{\partial^2 \mathcal{H}}{\partial z_i \partial v_j} & -\frac{\partial^2 \mathcal{H}}{\partial z_i \partial z_j} \\ \frac{\partial^2 \mathcal{H}}{\partial v_i \partial z_j} & -\frac{\partial^2 \mathcal{H}}{\partial v_i \partial z_j} \end{pmatrix} \Delta t \approx 1 \end{aligned} \tag{13}$$

The deterministic conditional can thus be written as:

$$q(\boldsymbol{z}_{t+\frac{1}{2}}, \boldsymbol{v}_{t+\frac{1}{2}}|\boldsymbol{z}_t, \boldsymbol{v}_t) \approx \delta\Big(\boldsymbol{z}_{t+\frac{1}{2}} - f_z(\boldsymbol{z}_t, \boldsymbol{v}_t), \boldsymbol{v}_{t+\frac{1}{2}} - f_v(\boldsymbol{z}_t, \boldsymbol{v}_t)\Big) \tag{14}$$

The posterior conserves probability mass over time. Let $q(\boldsymbol{z}_{t+1}|\boldsymbol{z}_{t+1/2})$ denote a trivial Dirac δ-function. Marginalizing $\boldsymbol{z}_{t+1/2}$ out gives the conditional of $\boldsymbol{z}_{t+1}$:

$$\begin{aligned} q(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{v}_t) &= \int q(\boldsymbol{z}_{t+1}|\boldsymbol{z}_{t+\frac{1}{2}}) q(\boldsymbol{z}_{t+\frac{1}{2}}|\boldsymbol{z}_t, \boldsymbol{v}_t)\, d\boldsymbol{z}_{t+\frac{1}{2}} \\ &= \int \delta\Big(\boldsymbol{z}_{t+1} - \boldsymbol{z}_{t+\frac{1}{2}}\Big)\delta\Big(\boldsymbol{z}_{t+\frac{1}{2}} - f_{\boldsymbol{z}}(\boldsymbol{z}_t, \boldsymbol{v}_t)\Big)\, d\boldsymbol{z}_{t+\frac{1}{2}} \\ &= \delta\Big(\boldsymbol{z}_{t+1} - f_{\boldsymbol{z}}(\boldsymbol{z}_t, \boldsymbol{v}_t)\Big) \end{aligned} \tag{15}$$

Ornstein-Uhlenbeck Process. The probabilistic step of Eq. (9) is given by the Ornstein–Uhlenbeck process which describes a noisy relaxation process, whereby a particle is disturbed with noise $\eta(t)$ and simultaneously relaxed to its mean position with friction coefficient γ:

$$\frac{d\boldsymbol{v}}{dt} = -\gamma\boldsymbol{v} + \sqrt{2m\gamma k_B \tau}\, \eta(t). \tag{16}$$

Discretizing over timesteps, the Gaussian noise yields a Gaussian transition update as:

$$\begin{aligned} q(\boldsymbol{v}_{t+1}|\boldsymbol{z}_t, \boldsymbol{v}_t) &= \int q(\boldsymbol{v}_{t+1}|\boldsymbol{v}_{t+\frac{1}{2}}) q(\boldsymbol{v}_{t+\frac{1}{2}}|\boldsymbol{z}_t, \boldsymbol{v}_t)\, d\boldsymbol{v}_{t+\frac{1}{2}} \\ &= \int \mathcal{N}\Big((1-\gamma)\boldsymbol{v}_{t+\frac{1}{2}}, 2m\gamma k_B \tau I\Big)\delta\Big(\boldsymbol{v}_{t+\frac{1}{2}} - f_{\boldsymbol{v}}(\boldsymbol{z}_t, \boldsymbol{v}_t)\Big)\, d\boldsymbol{v}_{t+\frac{1}{2}} \\ &= \mathcal{N}\Big((1-\gamma)f_{\boldsymbol{v}}(\boldsymbol{z}_t, \boldsymbol{v}_t), 2m\gamma k_B \tau I\Big) \end{aligned} \tag{17}$$

The re-parameterization trick (Kingma & Welling, 2013) is used to allow for differentiation through the Gaussian kernel. We alternate these two steps to compute the conditional update of the joint posterior $q(\boldsymbol{z}_{t+1}, \boldsymbol{v}_{t+1}|\boldsymbol{z}_t, \boldsymbol{v}_t)$.

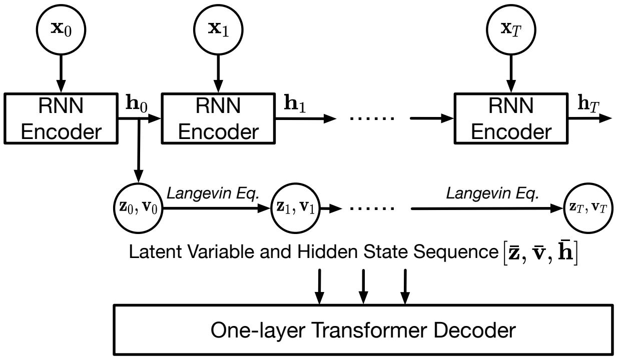
$\mathbf{x}_0$
$\mathbf{x}_1$
$\mathbf{x}_T$
RNN Encoder
$\mathbf{h}_0$
RNN Encoder
$\mathbf{h}_1$
......
RNN Encoder
$\mathbf{h}_T$
$\mathbf{z}_0, \mathbf{v}_0$
Langevin Eq.
$\mathbf{z}_1, \mathbf{v}_1$
......
Langevin Eq.
$\mathbf{z}_T, \mathbf{v}_T$
Latent Variable and Hidden State Sequence $[\bar{\mathbf{z}}, \bar{\mathbf{v}}, \bar{\mathbf{h}}]$
One-layer Transformer Decoder

Figure 1: Workflow of our method: the RNN encoder takes the spike data as input at every timestep and updates the hidden states $\boldsymbol{h}_t$, and the latent variables $\boldsymbol{z}_t, \boldsymbol{v}_t$ evolve in time according to the Langevin equation. Finally, the Transformer decoder predicts the firing rates from the entire sequence.

Architecture and Training Algorithm

Fig. 1 displays our model architecture. A recurrent encoder `GRU` (Chung et al., 2014) is used to encode the input sequence to a set of hidden states $\mathbf{h}_t = \texttt{GRU}(\mathbf{x}_{t-1}, \mathbf{h}_{t-1})$. The initial conditions for the latent variables $\mathbf{z}_0$ & $\mathbf{v}_0$ are inferred from $\mathbf{h}_0$, and then evolve forward in time according to both the deterministic and stochastic steps. The RNN encoder is included

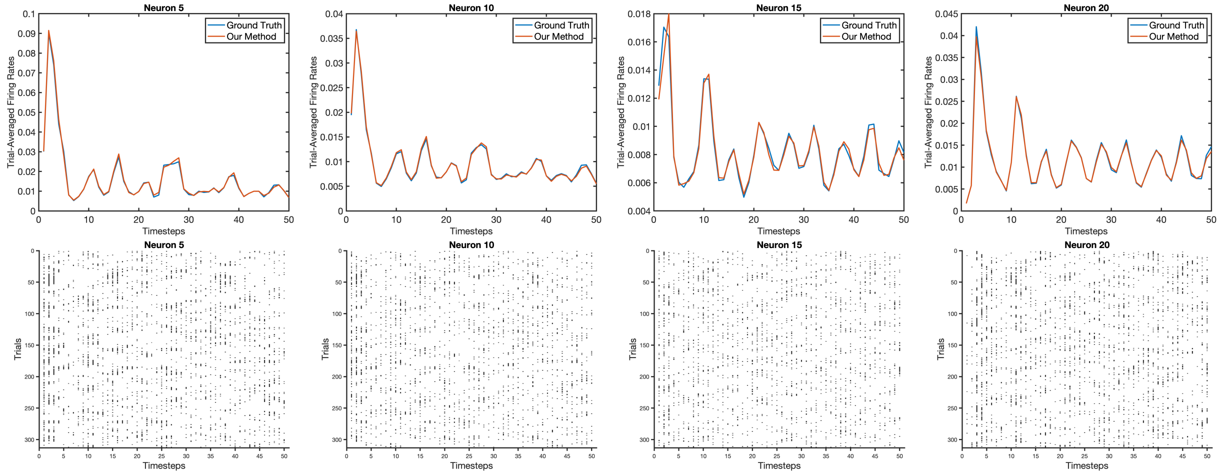

Figure 2: Trial-average firing rates (top) and the corresponding spike trains (bottom) of some neurons of Lorenz system.

Table 1: Results on `MC_Maze` and `MC_RTT` with the sampling frequency of 20 ms.

Methods	MC-Maze				MC-RTT		
	co-bps (↑)	vel R2 (↑)	psth R2 (↑)	fp-bps (↑)	co-bps (↑)	vel R2 (↑)	fp-bps (↑)
Smoothing (Yu et al., 2008)	0.2076	0.6111	-0.0005	–	0.1454	0.3875	–
GPFA (Yu et al., 2008)	0.2463	0.6613	0.5574	–	0.1769	0.5263	–
SLDS (Linderman et al., 2017)	0.2117	0.7944	0.4709	-0.1513	0.1662	0.5365	-0.0509
NDT (Ye & Pandarinath, 2021)	**0.3597**	0.8897	0.6172	0.2442	0.1643	0.6100	0.1200
AutoLFADS (Pandarinath et al., 2018b)	0.3554	0.8906	0.6002	**0.2454**	0.1976	0.6105	**0.1241**
MINT (Perkins et al., 2023)	0.3295	**0.9005**	**0.7474**	0.2076	**0.2008**	**0.6547**	0.1099
LangevinFlow	**0.3641**	**0.8940**	**0.6801**	**0.2573**	**0.2010**	**0.6652**	**0.1389**

```
Algorithm 1 Training algorithm of our Langevin flow.
Require: Recurrent encoder GRU, Transformer-based sequence decoder Transformer, linear mapping for latent
    variables $n$, input spike sequence $\bar{x}$, and posterior $q_\theta$.
repeat
    Initial hidden states: $h_0 = \mathrm{GRU}(x_0)$
    Initial latent variables: $[z_0, v_0] = n(h_0)$
    Time step counter: $i = 0$
    while $i \leq T - 1$ do
        Update position (deterministic step): $z_{i+1} = z_i + v_i$
        Update velocity (deterministic step):
            $v_{i+\frac{1}{2}} = v_i - \nabla_z U(z_i)/m$
        Update velocity (probabilistic step):
            $v_{i+1} = (1-\gamma) v_{i+\frac{1}{2}} + \sqrt{2m\gamma k_B \tau}\, \eta(i)$
        Update hidden states: $h_{i+1} = \mathrm{GRU}(x_{i+1}, h_i)$
        Concatenate variable sequences:
            $\bar{z} = [z_{0:i}, z_{i+1}],\ \bar{v} = [v_{0:i}, v_{i+1}],\ \bar{h} = [h_{0:i}, h_{i+1}]$
        Update time step counter: $i = i + 1$
    end while
    Predict firing rates: $\bar{r} = \mathrm{Transformer}(\bar{z}, \bar{v}, \bar{h})$
    Optimize the $\mathcal{L}^{\text{Poisson}}$ and $\mathcal{L}^{KL}$.
until converged
```

to model the short-range dependencies of neural data. After encoding input spikes and performing latent Langevin flow, all hidden states and latent variables are combined through a single Transformer (Vaswani et al., 2017) layer to predict the firing rates of the sequence: $\bar{\mathbf{r}} = \mathtt{Transformer}(\bar{\mathbf{z}}, \bar{\mathbf{v}}, \bar{\mathbf{h}})$. We use a Transformer for decoding because it can capture long-range interactions over time and allows for a more globally informed prediction of firing rates. The parameters of the GRU, Transformer, linear readout, and potential are then optimized to maximize the ELBO in Eq. (6). We summarize the training algorithm in Alg. 1.

Experiments

This section presents the experimental setup and the results of our method. We start with the setup of the experiments, discuss the results on the toy dataset of synthetic Lorenz attractor, and finally present the extensive evaluation of the Neural Latents Benchmark.

Setup

Baselines. On the synthetic Lorenz attractor dataset, we mainly compare with AutoLFADS (Pandarinath et al., 2018a) and NDT (Ye & Pandarinath, 2021), which are dedicated RNN and Transformer architectures designed for neural population modeling. On NLB, we further compare with a wide range of

Table 2: Results on `Area2_Bump` and `DMFC_RSG` with the sampling frequency of 20 ms.

Methods	Area2-Bump				DMFC-RSG			
	co-bps (↑)	vel R2 (↑)	psth R2 (↑)	fp-bps (↑)	co-bps (↑)	tp corr (↓)	psth R2 (↑)	fp-bps (↑)
Smoothing (Yu et al., 2008)	0.1529	0.5319	-0.1840	–	0.1183	-0.5115	0.2830	–
GPFA (Yu et al., 2008)	0.1791	0.6094	0.5998	–	0.1378	-0.5506	0.3180	–
SLDS (Linderman et al., 2017)	0.1816	0.6967	0.5200	0.0132	0.1575	-0.5997	0.5470	0.0374
NDT (Ye & Pandarinath, 2021)	0.2624	0.8623	0.6078	0.1459	0.1757	-0.6928	0.5477	0.1649
AutoLFADS (Pandarinath et al., 2018b)	0.2542	0.8565	0.6552	0.1423	0.1871	**-0.7819**	0.5903	0.1791
MINT (Perkins et al., 2023)	0.2718	0.8803	**0.9049**	0.1489	0.1824	-0.6995	**0.7014**	0.1647
LangevinFlow	**0.2881**	**0.8810**	0.7641	**0.1647**	**0.1904**	-0.5981	0.6079	**0.1945**

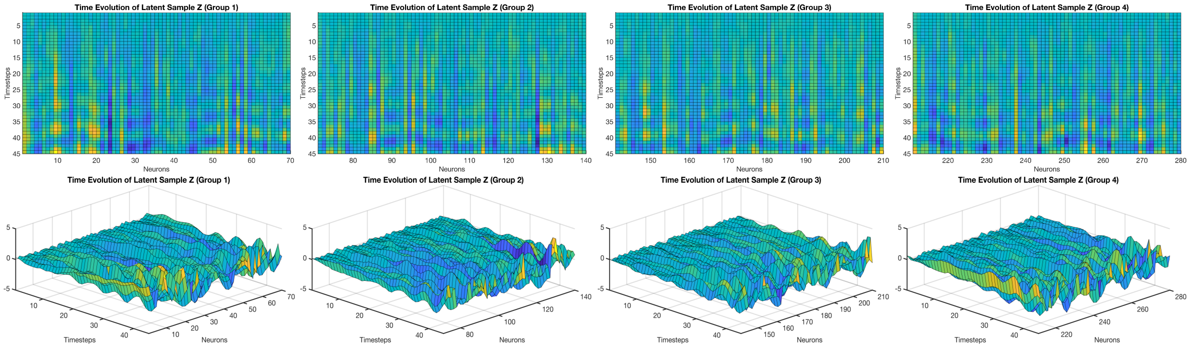

Figure 3: Spatiotemporal waves induced by our LangevinFlow in different views on `MC_Maze`. Here each group denotes an independent set of convolution channels.

competitive baselines on the public leaderboard [1].

Implementation Details. The Transformer decoder consists of only 1 self-attention layer with 4 attention heads. Another linear layer is used for reading out firing rates. For the Langevin equation, the mass m is set to an identity matrix, and both Boltzman constant k_B and temperature τ are set 1. The damping ratio γ is tuned for specific datasets but stays in the range of $[0.55, 0.8]$. For the potential, the latent code is first divided into 4 groups (*i.e.,* independent convolution channels), and we use a one-dimensional convolution layer of kernel size 7 with padding 3 and stride 1 for each group. We adopt a hyper-parameter λ to tune the strength of the KL penalty and add a scheduler to gradually increase the value so that the optimization does not quickly set the KL divergence to 0. As the observed spikes are assumed to be from a low-dimensional subspace, we use coordinated dropout (Keshtkaran & Pandarinath, 2019) to randomly drop input samples during the training, which enforces the model to learn the underlying latent structure shared across neurons.

Synthetic Lorenz Attractor

The Lorenz attractor is a 3D dynamical system where the dynamics are governed by three coupled non-linear equations:

$$\begin{aligned} \dot{y}_1 &= \sigma(y_2 - y_3), \\ \dot{y}_2 &= y_1(\rho - y_3)y_2, \\ \dot{y}_3 &= y_1 y_2 - \beta y_3 \end{aligned} \tag{18}$$

where σ, ρ, β are hyper-parameters. In line with Ye & Pandarinath (2021), we first simulate the 3D Lorenz attractor and then project the 3D states into a higher dimensionality using a random linear transform to form firing rates for a population of synthetic neurons. The spikes of each trial are sampled from the Poisson distribution with these firing rates. The evaluating methods are expected to infer the true firing rates of the Lorenz system from the synthetic spiking activity alone.

Table 3: R_2 of the firing rates on Lorenz Attractor.

	AutoLFADS	NDT	LangevinFlow
$R_2(\uparrow)$	0.921±0.005	0.934±0.004	**0.944±0.003**

Table 3 presents the results of R_2 correlation between the predicted firing rates and the ground truth. Our LangevinFlow outperforms the baselines and achieves a higher correlation score in predicting the firing rates, indicating that our model more accurately captures the underlying dynamical structure of the synthetic neural data. Fig. 2 compares the predicted trial-averaged firing rates and of several randomly selected neurons alongside their ground truth counterparts, as well as the corresponding spike trains. Our method closely recovers the general shape and amplitude of the firing rate curves, and also accurately reflects the temporal structure of spike trains.

Neural Latent Benchmark

The NLB (Pei et al., 2021) is a benchmark designed for evaluating unsupervised approaches that model neural popula-

[1] https://eval.ai/web/challenges/challenge-page/1256/leaderboard/

Table 4: Results on MC_Maze and MC_RTT with the sampling frequency of 5 ms.

Methods	MC-Maze				MC-RTT		
	co-bps (↑)	vel R2 (↑)	psth R2 (↑)	fp-bps (↑)	co-bps (↑)	vel R2 (↑)	fp-bps (↑)
Smoothing (Yu et al., 2008)	0.2109	0.6238	0.1853	–	0.1468	0.4142	–
GPFA (Yu et al., 2008)	0.1872	0.6399	0.5150	–	0.1548	0.5339	–
SLDS (Linderman et al., 2017)	0.2249	0.7947	0.5330	-0.1513	0.1649	0.5206	0.0620
NDT (Ye & Pandarinath, 2021)	0.3229	0.8862	0.5308	0.2206	0.1749	0.5656	0.0970
AutoLFADS (Pandarinath et al., 2018b)	0.3364	0.9097	0.6360	0.2349	0.1868	0.6167	0.1213
MINT (Perkins et al., 2023)	0.3304	**0.9121**	**0.7496**	0.2076	**0.2014**	**0.6559**	0.1099
LangevinFlow	**0.3624**	0.7867	0.5515	**0.2556**	0.1900	0.4748	**0.1300**

Table 5: Results on Area2_Bump and DMFC_RSG with the sampling frequency of 5 ms.

Methods	Area2-Bump				DMFC-RSG			
	co-bps (↑)	vel R2 (↑)	psth R2 (↑)	fp-bps (↑)	co-bps (↑)	tp corr (↓)	psth R2 (↑)	fp-bps (↑)
Smoothing (Yu et al., 2008)	0.1544	0.5736	0.2084	–	0.1202	-0.5139	0.2993	–
GPFA (Yu et al., 2008)	0.1680	0.5975	0.5289	–	0.1176	-0.3763	0.2142	–
SLDS (Linderman et al., 2017)	0.1960	0.7385	0.5740	0.0242	0.1243	-0.5412	0.3372	-0.0418
NDT (Ye & Pandarinath, 2021)	0.2623	0.8672	0.6619	0.1184	0.1720	-0.5624	0.4377	0.1404
AutoLFADS (Pandarinath et al., 2018b)	0.2569	0.8492	0.6318	0.1505	0.1829	**-0.8248**	0.6359	**0.1844**
MINT (Perkins et al., 2023)	0.2735	**0.8877**	**0.9135**	0.1483	0.1821	-0.6929	**0.7013**	0.1650
LangevinFlow	**0.2772**	0.8580	0.7567	**0.1526**	**0.1841**	-0.5466	0.6092	0.1689

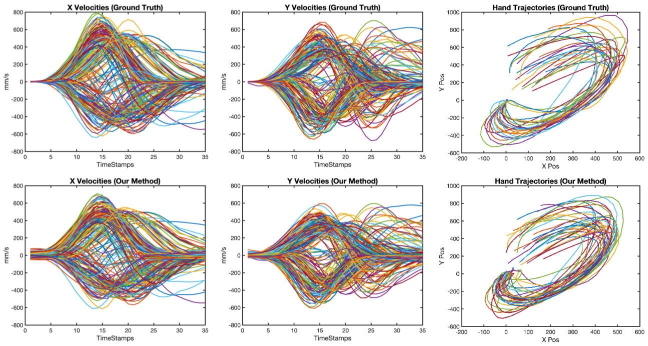

Figure 4: Kinematics (hand velocities and trajectories) of the ground truth and predicted by our method on Area2_Bump.

tion activities. This benchmark provides four curated neurophysiological datasets from monkeys that span motor, sensory, and cognitive brain regions, with behaviors that vary from pre-planned, stereotyped movements to those in which sensory input must be dynamically integrated. The primary metric co-smoothing (Macke et al., 2011) evaluates the normalized log-likelihood of held-out neuronal activity prediction, while the secondary metrics can include behavior decoding accuracy, match to PSTH, or log-likelihood of forward predictions. Two sampling frequencies (5ms and 20ms) are pre-defined to obtain datasets with different sequence lengths.

20 ms Results. In Table 1 and 2, we see that our LangevinFlow achieves state of the art on the likelihood of held-out neurons (co-smoothing bits per spike), as well as forward prediction bits per spike. The model also compares very favorably in terms of the behavioral metrics such as hand-velocity regression. The model is not state of the art on PSTH R^2, which may be expected given the known trade-off between the co-bps metric and the performance on trial-averaged PSTH correlation metric. The overall performance across multiple metrics underscores its robustness in capturing neural dynamics.

5 ms Results. Table 4 and 5 report the evaluation results on NLB with the sampling frequency of 5 ms. The results at this higher temporal resolution are very coherent with those of 20 ms. Our LangevinFlow maintains impressive performance, achieving strong likelihood scores on held-out neurons and forward predictions on most datasets. This consistency across different sampling frequencies confirms the model's ability to adapt to varying temporal granularities, which is critical for capturing the fine-scale dynamics present in neural data.

Spatiotemporal Wave Dynamics. Fig. 3 displays the smooth spatiotemporal waves induced by our coupled oscillator potential. We can see that the latent variables in different convolution groups exhibit clear but distinct wave patterns, reminiscent of traveling waves observed in cortical activity (Ermentrout & Kleinfeld, 2001; Muller et al., 2014). Such wave dynamics are thought to play several key computational roles in neuroscience. For example, traveling waves have been proposed to facilitate the integration of information over distributed neural populations, serving as a mechanism for coordinating activity across different brain regions (Buzsaki, 2006; Miller et al., 2009; Muller et al., 2018). They can help synchronize the timing of neural firing, thereby enhancing signal propagation and ensuring that information is efficiently routed and integrated. Moreover, wave dynamics may support processes such as working memory, decision-making, preditive decoding, and sensorimotor integration (Sato et al., 2012; Engel et al., 2013; Besserve et al., 2015; Alamia & VanRullen, 2019; Friston, 2019). In our model, the emergence of these wave patterns not only reflects the inherent oscillatory dynamics of the neural data, but also suggests that our coupled oscillator potential may be capturing similar computational principles, contributing to the robust performance of our approach.

Kinematics Visualization. Fig. 4 illustrates the time evolution of key kinematic variables on the `Area2_Bump` dataset, including hand X and Y velocities, as well as the overall hand trajectories. The high behavior decoding accuracy of our model is evident here: linear regression models fitted on predicted firing rates yield kinematic outputs that closely match the ground truth. These results validate the accuracy of the neural activity reconstruction, demonstrating the practical utility of our approach in decoding behaviorally relevant signals.

Table 6: Results of ablation studies with the sampling frequency of 20 ms on `MC_Maze` (top) and `Area2_Bump` (bottom).

	co-bps (↑)	vel R2 (↑)	psth R2 (↑)	fp-bps (↑)
LangevinFlow	**0.3641**	0.8940	**0.6801**	**0.2573**
Baseline 1	0.3572	0.8893	0.6683	0.2419
Baseline 2	0.3328	0.8579	0.6812	0.2549
Baseline 3	0.3441	0.8109	0.4684	0.2506
Baseline 4	0.3612	**0.9005**	0.6743	0.2469
Baseline 5	0.3586	0.8932	0.6881	0.2351

	co-bps (↑)	vel R2 (↑)	psth R2 (↑)	fp-bps (↑)
LangevinFlow	**0.2881**	**0.8810**	0.7641	**0.1647**
Baseline 1	0.2795	0.8725	0.7386	0.1549
Baseline 2	0.2679	0.8641	0.7013	0.1488
Baseline 3	0.2838	0.8552	0.7165	0.1498
Baseline 4	0.2800	0.8739	**0.7803**	0.1631
Baseline 5	0.2843	0.8614	0.6994	0.1596

Ablation Studies. Finally, we designed a number of baselines and performed ablations to understand the role each component of our LangevinFlow plays on overall performance. Specifically, we considered the following model variants:

- *Baseline 1*: a linear decoder in place of the Transformer.
- *Baseline 2*: a linear encoder with no hidden states.
- *Baseline 3*: a model without Langevin dynamics relying solely on hidden state dynamics.
- *Baseline 4*: a variant in which the oscillator potential also couples latent variables to input spikes. Explicitly: $U(\boldsymbol{z},\boldsymbol{x}) = \boldsymbol{z}^T \frac{\boldsymbol{W}_z}{||\boldsymbol{W}_z||_2} \boldsymbol{z} + \boldsymbol{z}^T \boldsymbol{W}_x \boldsymbol{x}$.
- *Baseline 5*: a version using first-order dynamics instead of Langevin dynamics. Explicitly: $\boldsymbol{z}_{t+1} = \boldsymbol{z}_t - \nabla_{\boldsymbol{z}_t} U(\boldsymbol{z}_t)$.

Table 6 shows the results on `MC_Maze` and `Area2_Bump`. In *Baseline 1*, we replaced the Transformer decoder with a linear decoder. Compared to LangevinFlow, this variant has slightly lower co-smoothing (co-bps) and forward prediction (fp-bps) scores, which indicate the importance of the Transformer in capturing global interactions across the entire latent sequence and in refining firing rate predictions. The global attention mechanism appears to integrate information more effectively than a simpler linear mapping.

Baseline 2 removes the hidden states from the encoder by replacing the recurrent network with a linear encoder. This modification leads to a noticeable drop in performance, particularly in the co-bps and velocity R2 scores. This suggests that the local temporal dependencies captured by recurrent hidden states are essential for modeling the short-range dynamics present in the neural spike activity.

Baseline 3 completely omits Langevin dynamics and relies solely on hidden state dynamics. This modification results in a marked performance drop, especially evident in the significant drop in PSTH R2 on `MC_Maze`. This decline emphasizes the crucial role of incorporating Langevin dynamics with a learned potential, which represents intrinsic autonomous processes and facilitates the emergence of oscillations. The Langevin dynamics are thus expected to help the model capture the underlying dynamical system more faithfully.

In *Baseline 4*, we augment the oscillator potential by incorporating the input spiking signal. This modification does not provide a substantial benefit to the performance and in some cases slightly underperforms the original model. This result suggests that the learned potential function in its original formulation is already capturing the necessary dependencies.

Finally, *Baseline 5* substitutes the second-order Langevin dynamics with a simpler first-order update rule. The observed performance drop in several metrics confirms that the second-order Langevin dynamics – featuring terms for inertia and damping – is more effective in modeling the neural dynamics. The richer dynamics afforded by the second-order formulation appear to better capture both the smooth evolution and the inherent variability of the underlying latent factors.

Conclusions

This paper presents LangevinFlow, a sequential variational autoencoder whose latent dynamics are governed by underdamped Langevin equations. By embedding physically grounded stochastic processes and coupled oscillatory behavior into the latent space, our framework offers a powerful avenue for modeling complex neural population activity. We anticipate that these ideas will inspire further exploration of physics-informed inductive biases in neural latent variable modeling, paving the way for even richer and more interpretable dynamical systems approaches.

Limitations and Future Work. While our framework was shown to yield very promising results, our proposed Langevin dynamics with the present potential function operate in a largely autonomous manner. This formulation seemed to work better than an input-dependent potential in ablation studies; however, adding more input dependence to this potential should intuitively help Langevin dynamics better account for external influences. In future work, exploring more complex input-dependent potential functions could likely yield significant benefits and are a promising new avenue for research uniquely enabled by our LangevinFlow framework.

References

Alamia, A., & VanRullen, R. (2019). Alpha oscillations and traveling waves: Signatures of predictive coding? *PLoS Biology*, *17*(10), e3000487.

Azabou, M., Arora, V., Ganesh, V., Mao, X., Nachimuthu, S., Mendelson, M., . . . Dyer, E. (2024). A unified, scalable framework for neural population decoding. *Advances in Neural Information Processing Systems*, *36*.

Besserve, M., Lowe, S. C., Logothetis, N. K., Schölkopf, B., & Panzeri, S. (2015). Shifts of gamma phase across primary visual cortical sites reflect dynamic stimulus-modulated information transfer. *PLoS biology*, *13*(9), e1002257.

Buzsaki, G. (2006). *Rhythms of the brain*. Oxford university press.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Churchland, M. M., Byron, M. Y., Ryu, S. I., Santhanam, G., & Shenoy, K. V. (2006). Neural variability in premotor cortex provides a signature of motor preparation. *Journal of Neuroscience*, *26*(14), 3697–3712.

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., & Shenoy, K. V. (2012). Neural population dynamics during reaching. *Nature*, *487*(7405), 51–56.

Cohen, M. R., & Kohn, A. (2011). Measuring and interpreting neuronal correlations. *Nature neuroscience*, *14*(7), 811–819.

Dabagia, M., Kording, K. P., & Dyer, E. L. (2023). Aligning latent representations of neural activity. *Nature Biomedical Engineering*, *7*(4), 337–343.

Diamant, N., & Bortoff, A. (1969, February). Nature of the intestinal low-wave frequency gradient. *American Journal of Physiology-Legacy Content*, *216*(2), 301–307. doi: 10.1152/ajplegacy.1969.216.2.301

Duncker, L., Bohner, G., Boussard, J., & Sahani, M. (2019). Learning interpretable continuous-time models of latent stochastic dynamical systems. In *International conference on machine learning* (pp. 1726–1734).

Duncker, L., & Sahani, M. (2018). Temporal alignment and latent gaussian process factor inference in population spike trains. *Advances in neural information processing systems*, *31*.

Ecker, A. S., Berens, P., Keliris, G. A., Bethge, M., Logothetis, N. K., & Tolias, A. S. (2010). Decorrelated neuronal firing in cortical microcircuits. *science*, *327*(5965), 584–587.

Engel, A. K., Gerloff, C., Hilgetag, C. C., & Nolte, G. (2013). Intrinsic coupling modes: multiscale interactions in ongoing brain activity. *Neuron*, *80*(4), 867–886.

Ermentrout, G. B., & Kleinfeld, D. (2001). Traveling electrical waves in cortex: insights from phase dynamics and speculation on a computational role. *Neuron*, *29*(1), 33–44. doi: https://doi.org/10.1016/S0896-6273(01)00178-7

Ermentrout, G. B., & Kopell, N. (1984). Frequency plateaus in a chain of weakly coupled oscillators, i. *SIAM journal on Mathematical Analysis*, *15*(2), 215–237.

Friston, K. J. (2019). Waves of prediction. *PLoS biology*, *17*(10), e3000426.

Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A., & Miller, L. E. (2020). Long-term stability of cortical population dynamics underlying consistent behavior. *Nature neuroscience*, *23*(2), 260–270.

Gallego, J. A., Perich, M. G., Miller, L. E., & Solla, S. A. (2017). Neural manifolds for the control of movement. *Neuron*, *94*(5), 978-984. doi: https://doi.org/10.1016/j.neuron.2017.05.025

Gallego, J. A., Perich, M. G., Naufel, S. N., Ethier, C., Solla, S. A., & Miller, L. E. (2018). Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature communications*, *9*(1), 4233.

Gao, Y., Archer, E. W., Paninski, L., & Cunningham, J. P. (2016). Linear dynamical neural population models through nonlinear embeddings. *Advances in neural information processing systems*, *29*.

Genkin, M., Shenoy, K. V., Chandrasekaran, C., & Engel, T. A. (2023). The dynamics and geometry of choice in premotor cortex. *bioRxiv*. doi: 10.1101/2023.07.22.550183

Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., … others (2022). Perceiver io: A general architecture for structured inputs & outputs. In *International conference on learning representations.*

Kao, J. C., Nuyujukian, P., Ryu, S. I., Churchland, M. M., Cunningham, J. P., & Shenoy, K. V. (2015). Single-trial dynamics of motor cortex and their applications to brain-machine interfaces. *Nature communications*, *6*(1), 7759.

Karpowicz, B. M., Ali, Y. H., Wimalasena, L. N., Sedler, A. R., Keshtkaran, M. R., Bodkin, K., … Pandarinath, C. (2022). Stabilizing brain-computer interfaces through alignment of latent dynamics. *BioRxiv*, 2022–04.

Keshtkaran, M. R., & Pandarinath, C. (2019). Enabling hyperparameter optimization in sequential autoencoders for spiking neural data. *NeurIPS*.

Kingma, D. P., & Welling, M. (2013). *Auto-encoding variational bayes.*

Linderman, S., Johnson, M., Miller, A., Adams, R., Blei, D., & Paninski, L. (2017). Bayesian learning and inference in recurrent switching linear dynamical systems. In *Aistats.*

Macke, J. H., Buesing, L., Cunningham, J. P., Yu, B. M., Shenoy, K. V., & Sahani, M. (2011). Empirical models of spiking in neural populations. *NeurIPS*, *24*.

Miller, K. J., Sorensen, L. B., Ojemann, J. G., & Den Nijs, M. (2009). Power-law scaling in the brain surface electric potential. *PLoS computational biology*, *5*(12), e1000609.

Muller, L., Chavane, F., Reynolds, J., & Sejnowski, T. J. (2018). Cortical travelling waves: mechanisms and computational principles. *Nature Reviews Neuroscience*, *19*(5), 255–268.

Muller, L., Reynaud, A., Chavane, F., & Destexhe, A. (2014). The stimulus-evoked population response in visual cortex of awake monkey is a propagating wave. *Nature communications*, *5*(1), 3675.

Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., … others (2018a). Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*.

Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., … others (2018b). Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*.

Pei, F., Ye, J., Zoltowski, D., Wu, A., Chowdhury, R. H., Sohn, H., … others (2021). Neural latents benchmark'21: evaluating latent variable models of neural population activity. *NeurIPS*.

Perkins, S. M., Cunningham, J. P., Wang, Q., & Churchland, M. M. (2023). Simple decoding of behavior from a complicated neural manifold. *BioRxiv*, 2023–04.

Sato, T. K., Nauhaus, I., & Carandini, M. (2012). Traveling waves in visual cortex. *Neuron*, *75*(2), 218–229.

Saxena, S., & Cunningham, J. P. (2019). Towards the neural population doctrine. *Current opinion in neurobiology*, *55*, 103–111.

Shenoy, K. V., Sahani, M., & Churchland, M. M. (2013). Cortical control of arm movements: a dynamical systems perspective. *Annual review of neuroscience*, *36*(1), 337–359.

Stevenson, I. H., & Kording, K. P. (2011). How advances in neural recording affect data analysis. *Nature neuroscience*, *14*(2), 139–142.

Sussillo, D., Jozefowicz, R., Abbott, L. F., & Pandarinath, C. (2016). *Lfads - latent factor analysis via dynamical systems.*

Sussillo, D., Stavisky, S. D., Kao, J. C., Ryu, S. I., & Shenoy, K. V. (2016). Making brain–machine interfaces robust to future neural variability. *Nature communications*, *7*(1), 13749.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., … Polosukhin, I. (2017). Attention is all you need. In *Neurips.*

Vyas, S., Golub, M. D., Sussillo, D., & Shenoy, K. V. (2020). Computation through neural population dynamics. *Annual review of neuroscience*, *43*(1), 249–275.

Wu, A., Roy, N. A., Keeley, S., & Pillow, J. W. (2017). Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems*, *30*.

Ye, J., Collinger, J., Wehbe, L., & Gaunt, R. (2024). Neural data transformer 2: multi-context pretraining for neural spiking activity. *Advances in Neural Information Processing Systems*, *36*.

Ye, J., & Pandarinath, C. (2021). Representation learning for neural population activity with neural data transformers. *Neurons, Behavior, Data analysis, and Theory*.

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S., Shenoy, K. V., & Sahani, M. (2008). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *NeurIPS*.

Zhao, Y., & Park, I. M. (2016). Interpretable nonlinear dynamic modeling of neural trajectories. *Advances in neural information processing systems*, *29*.

Zhao, Y., & Park, I. M. (2017). Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *Neural computation*, *29*(5), 1293–1316.