CELL-CLASS NEURAL ORDINARY DIFFERENTIAL EQUATION MODELS FOR PARSING BIOLOGICALLY-CONSTRAINED CONTRIBUTIONS TO NEURAL DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding how populations of individual neurons interact to shape the overall dynamics of neural activity is a central question in computational and systems neuroscience. Recent work has shown that neural ordinary differential equation (NODE) models are able to model neural activity dynamics with high accuracy and interpretability of the underlying dynamics. However, existing NODE models treat all neurons as part of a homogeneous group, preventing understanding of how underlying neural populations (e.g., excitatory and inhibitory cell classes) contribute to the overall dynamics. Here, we introduce Excitatory-Inhibitory NODE (EI-NODE) models. These models A) decompose the overall dynamics into components specific to each population, allowing understanding of each population's interactions with one another; and B) provide biological constraints on the contributions of excitatory and inhibitory populations towards the dynamics, using a variant of monotonic neural networks. Using both synthetic data and recorded neural activity data during a naturalistic climbing task with optogenetic perturbations, we show that EI-NODE models can provide equivalent or greater accuracy in dynamics modeling compared to alternative models, while enabling a newfound biologically constrained understanding of neural populations' interactions and roles in the underlying dynamics.

1 INTRODUCTION

Recent advances in large-scale neural recordings reveal rich, high-dimensional population activity that invites dynamical systems explanations (Jun et al., 2017; Stirman et al., 2016; Ahrens et al., 2013; Sauerbrei et al., 2020; Itokazu et al., 2018). Latent dynamical models capture these data via low-dimensional states whose temporal evolution generates observed activity, achieving strong denoising and alignment with behavior (Macke et al., 2011; Petreska et al., 2011; Pandarinath et al., 2018).

Neural ordinary differential equation models (Neural ODEs or NODEs), which model the change of dynamics over time with flexible neural networks, have recently emerged as a promising approach towards modeling the latent dynamics of neural population activity. NODEs have been shown to have excellent accuracy, predicting neural activity data more accurately than recurrent neural network (RNN) models (Kim et al., 2021). They have also demonstrated accurate modeling of low-dimensional dynamics within a low-dimensional space, whereas RNN models require a greater dimensionality (due to their lower dynamical flexibility) to accurately capture low-dimensional dynamics (Sedler et al., 2023). NODEs also can provide a particularly interpretable view of the underlying dynamics, as flow fields can easily be drawn in the low-dimensional space demonstrating how neural activity evolves over time, as was recently done to discover neural underpinnings of decision commitment (Luo et al., 2025).

Despite great promise, there are two fundamental limitations of existing NODE models, especially relating to modeling widely emerging datasets with multiple recorded neural populations (e.g. brain regions and cell types). First, there has been great interest in understanding and modeling the roles

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

of, and interactions between, different neural populations (Keeley et al., 2020; Perich et al., 2020; Liu et al., 2025; Dowling and Savin; Cunningham and Yu, 2014; Li et al., 2024; Koukuntla et al., 2024; Jha et al., 2025; Gokcen et al., 2022; 2023). This is evidenced by recent work within other dynamics architectures, for example linear (Semedo et al., 2014), switching linear (Glaser et al., 2020), and switching nonlinear dynamical systems (Karniol-Tambour et al., 2022), to decompose the overall latent dynamics into the sum of the influences of different populations. It is not possible within current NODE models to understand how populations interact to drive the overall neural dynamics of the system. Second, biological neural circuits comprise distinct excitatory (E) and inhibitory (I) cell populations with sign-constrained interactions (Dale's law). Unlike dynamics architectures like RNNs (Song et al., 2016) and linear dynamical systems (Jha et al., 2024), current NODE models don't enable incorporating these biological constraints. This omission prevents constraining the dynamics of models (which are often poorly constrained (Prinz et al., 2004; Das and Fiete, 2020; Beiran and Litwin-Kumar, 2024)) toward biological solutions, and obscures how distinct cell classes contribute to circuit computations.

To tackle these challenges, we developed Excitatory-Inhibitory NODE (EI-NODE) models. EI-NODE assigns separate latent subspaces to E and I populations and constrains cross-population influences in this latent space to respect Dale's law so that E (and I) neurons only excite (and inhibit). To do so, we develop a dynamics parameterization that builds upon monotonic neural networks (Daniels and Velikova, 2010; Sartor et al., 2025), which enables the dynamics to retain the flexibility inherent to NODE models, while now respecting biological constraints. After formulating the model (Section 2), we evaluate our model on synthetic data generated from a winner-take-all decision-making circuit (Section 3), and naturalistic mouse climbing data recorded in primary (M1) and secondary (M2) motor areas, including with optogenetic perturbations (Section 4). We demonstrate how EI-NODE models enable interpreting how individual populations contribute to the overall dynamics (e.g. within flow fields), and how they interact with each other over time, which is not possible within standard NODE models. Crucially, this added biological interpretability does not come at the cost of prediction performance versus competing models. Together, our findings suggest a path toward simple, accurate, and biologically grounded latent dynamics that elucidate computation in neural circuits.

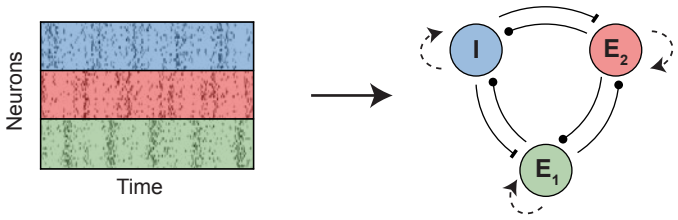

Figure 1: *EI-NODE Schematic*. (Left) Raster plots of neural firing rates over time for three example populations. (Right) A schematic of our EI-NODE model is trained to infer population interactions from neural activity for both E and I populations, with positive (dot), negative (line), and unconstrained (dashed arrow) interactions.

2 Excitatory-Inhibitory NODEs (EI-NODE)

2.1 Background: Neural Ordinary Differential Equations (NODEs)

NODEs model the dynamics of a variable z at time t as:

$$z_t \;=\; z_{t-1} \;+\; \int_{t-1}^{t} f_\theta\big(z_{t-1}\big)\, dt \tag{1}$$

where f_θ is a neural network parameterizing the latent dynamics vector field, and can be interpreted as the rate of change of z over time, $\frac{dz}{dt}$. Critically, this formulation provides a skip connection from the previous state, z_{t-1}, focusing f_θ's full expressivity towards dynamics.

2.2 BACKGROUND: NODE-BASED SEQUENTIAL AUTOENCODERS

Here, as in past work (Sedler et al., 2023), we model the high-dimensional neural activity as having a lower-dimensional latent representation, and the NODE dynamics occur within this latent space. This framework has been referred to as a NODE-based sequential autoencoder (Sedler et al., 2023), although we will just refer to these models as NODEs for brevity.

Let $s_t \in \mathbb{R}^N$ denote the observed binned spike counts of N neurons at time t. Predicted firing rates $\hat{x_t}$ are mapped from L latents $z_t \in \mathbb{R}^L$ with weights W_{dec}, biases b_{dec}, and an activation function ψ (typically either an exponential or softplus function) according to:

$$\hat{x_t} \;=\; \psi\big(W_{\text{dec}}\, z_t \;+\; b_{\text{dec}}\big), \qquad \hat{x_t} \in \mathbb{R}^N > 0. \tag{2}$$

Neural dynamics evolve in this lower dimensional state according to the dynamics specified in equation 1. The dynamics evolve in a deterministic manner from a learned initial latent state (the initial condition), z_0. For each sequence (e.g. trial) of data that is modeled, a separate initial condition is learned. We learn initial conditions by passing the high-dimensional neural activity through a Bidirectional Gated Recurrent Unit (BiGRU)-based encoder. More precisely, spike counts s over time T are input into a BiGRU with learned weights ϕ, and whose final hidden states are concatenated $h_T = [h_{T_{\text{forward}}}, h_{T_{\text{backward}}}]$ and linearly mapped to the initial latent z_0 using encoder weights W_{enc} and biases b_{enc}.

$$h \;=\; \text{BiGRU}_\phi\big(s_{1:T}\big), \qquad z_0 \;=\; W_{\text{enc}}\, h_T \;+\; b_{\text{enc}} \tag{3}$$

2.3 THE EI-NODE MODEL

Given simultaneously recorded E and I neural populations, our objective is to disentangle the roles of distinct cell types in a population. To accomplish this we modify the standard NODE model in three ways: (1) We define a distinct set of latent variables for the activity of each cell class; (2) For a given cell class, the dynamics of its latent variables are defined to be the summation of all population interactions; and (3) We constrain the interactions to obey functional properties of these cell types. We refer to the resulting models as EI-NODE models.

Cell-Type Latent Variables Definition. For K simultaneously recorded populations, we define a latent state ($z \in \mathbb{R}^L$) composed of K distinct sets of latents, where the latents of population j (z^j) for $j = 1, 2, ..., K$ can have any dimensionality D_j such that $L = D_1 + ... + D_K$. That is, $z = [z^1; \ldots; z^K]$ with $z^j \in \mathbb{R}^{D_j}$.

Dynamics Definition. We model the latent dynamics (i.e., the latent vector field) as the sum of population interactions plus a self-term. Therefore, the total change for population state z^j is defined by:

$$\tau_j \frac{dz^j(t)}{dt} = f_\theta^{jj}(z^j(t)) + \sum_{k \neq j} \tilde{f}_\theta^{jk}(z^k(t)) \tag{4}$$

where $\tau_j \in \mathbb{R}^{D_j}$ is a vector of learned latent time constants, f_θ^{jj} is a learned intrinsic self-term, and $\tilde{f}_\theta^{jk}$ is a learned influence from population k onto population j. We define each $\tilde{f}_\theta^{jk}$ to be a multilayer perceptron (MLP) and f_θ^{jj} to be a linear layer that acts independently on each latent within the jth population (analogous to the diagonal terms in E/I constrained linear dynamical systems (Jha et al., 2024)).

Constraining Dynamics to have Functional Properties. We constrain the latent population interactions such that it reflects the biological properties of its corresponding cell types: latents associated with E neurons will have a positive and monotonically increasing effect, while latents associated with I neurons will have a negative and monotonically decreasing effect. To ensure the learned interactions are monotonic, we explicitly define $\tilde{f}_\theta^{jk}$ to be a monotonic MLP (Daniels and Velikova, 2010; Sartor et al., 2025) (See details in Appendix A.2). Thus, an increasing (decreasing) input to $\tilde{f}_\theta^{jk}$ will lead to an increasing (decreasing) output. Since latent variables are not strictly constrained to be positive or negative, an excitatory latent can have an inhibitory effect if it has a negative value and is multiplied by its positive weight. This is a current limitation of applying Dale's law within other

dynamical systems frameworks. To allow for maximum interpretability, we constrain all weights to be nonnegative by passing the output of our monotonic neural network through a $\mathrm{softplus}$ activation function if it is excitatory, $\tilde{f}_\theta^{IE}$, to ensure the output is positive. When modeling the inhibitory interactions, $\tilde{f}_\theta^{EI}$, we simply take the negative of this approach to ensure a monotonically decreasing function. Specifically, $f_{IE} = \mathrm{softplus}(\tilde{f}_\theta^{IE})$ and $f_{EI} = -\mathrm{softplus}(\tilde{f}_\theta^{EI})$.

Mapping to the High-Dimensional Neural Space. In order for latent groups to be interpreted as corresponding to prespecified neural populations, we have specified that latent groups (e.g., z^E vs. z^I) be constrained to map onto their corresponding pre-specified neuron subsets (e.g. known E and I neural populations). We can thus write the readout as:

$$\hat{X}_t \;=\; \mathrm{softplus}\big(W_{\mathrm{dec}}\, z_t \;+\; b_{\mathrm{dec}}\big) \qquad W_{\mathrm{dec}} \;\leftarrow\; W_{\mathrm{dec}} \odot M_{\mathrm{pop}} \tag{5}$$

where M_{pop} is a block-diagonal binary mask mapping latents to their corresponding neurons. We note that in some experiments below, we relax this mask to test the ability of the model to learn neuron types without prespecifying them all (Section 3.3).

To ensure that the Dale's law constraints remain present in the high-dimensional neural activity space—e.g. all E neurons (not just E latents) have positive influences, we constrain all weights of W_{dec} to be nonnegative, as in past work (Jha et al., 2024).

We also note that we tested more flexible nonlinear readouts within our initial experiments on simulated data below, and saw similar performance—thus, we use a linear readout for simplicity.

2.4 Model Optimization and Baselines

We minimize the Poisson negative log-likelihood over T time steps and N neurons between predicted rates $\hat{X} = [\hat{y}_1, ..., \hat{y}_N]^T$, and observed spiking activity $S = [s_1, ..., s_N]^T$:

$$\mathcal{L}_\theta(S, \hat{X}) = -\frac{1}{TN} \sum_{t=1}^{T} \sum_{n=1}^{N} \log \mathrm{Poisson}\left(s_{t,n} \mid \hat{x}_{t,n}\right) \tag{6}$$

The results in this paper are shown for the discrete-time analogs of equation 1 and equation 4, $z_t = z_{t-1} + \Delta t\, f_\theta(z_{t-1})$, with $\Delta t = 0.1$. Discretization of the model allows for faster training and has been shown to produce equivalent results to the continuous formulation introduced by Chen et al. (2018) (Sedler et al., 2023; Kim et al., 2023).

Models were trained to minimize equation 6 using mini-batch Stochastic Gradient Descent (SGD) and an AdamW optimizer. The gradients of the loss with respect to the parameters θ were computed using the discrete adjoint sensitivity (i.e., standard backpropagation through time). Further training details are described in Appendix B.

We compare the EI-NODE model to several dynamics modeling approaches: 1) standard nonlinear NODE dynamics, to determine whether the cell-type-based constraints that facilitate interpretability impede performance accuracy; 2) cell-type constrained linear dynamics within the NODE framework, with the purpose of determining the importance of nonlinear dynamics, as cell-type constraints have previously been implemented within linear models (Jha et al., 2024); and 3) cell-type constrained RNN dynamics, as cell-type constraints have previously been implemented within RNNs (Song et al., 2016). For completeness, in Appendix E, we also compare against 4) unconstrained linear dynamics and 5) unconstrained RNN dynamics. Details of all these models are described in Appendix C.

3 Application to Simulated Neural Data

To demonstrate the feasibility of our method, we first validated its ability to recover known, interpretable circuit dynamics on synthetic data from a canonical decision-making circuit (Wang, 2002). Briefly, this dataset consists of two excitatory neural populations (E1 and E2), each containing 240 neurons, and one inhibitory population containing 400 neurons. Together, these populations form a decision-making model that achieves winner-take-all competition between the two E populations

based on noisy sensory input and feedback from the I population (Fig. 2A). This is a biophysically realistic network model that produces spiking activity across these three populations based on their interactions. More details are provided in Appendix B.1.

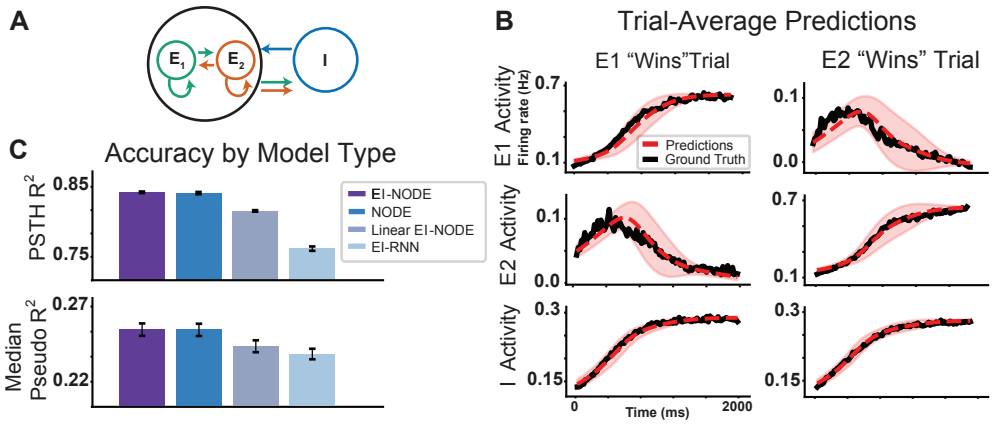

Figure 2: *Simulated decision-making data predictions*. **A)** Schematic of the synthetic dataset model (Wang, 2002; Stimberg et al., 2019), where spiking activity is generated from a bio-physically realistic spiking network consisting of two E populations and one I population. **B)** Trial-average predictions against ground truth firing rates for each population average. **C)** Prediction accuracy compared across model types (n=5). (Top) the average R^2 of neurons' PSTHs (averaged across trials) across all populations. (Bottom) The median pseudo-R^2 of individual neurons' spike count predictions. See appendix D for details about these metrics.

3.1 Results: Prediction Accuracy

We found that our EI-NODE model could make accurate predictions of the trajectories of neural activity over time (Fig. 2B). With a latent dimensionality of three (one latent per population, $z \in \mathbb{R}^3$), we quantitatively compared the EI-NODE model to several approaches of dynamics modeling based on their ability to reconstruct neural activity trajectories (Fig. 2C and Appendix E.2.3). First, the EI-NODE model performs equally with the NODE model, validating that introducing dynamical constraints does not hinder reconstruction performance while also adding the benefit of interpretability. Second, it also outperforms the Linear EI-NODE model, as expected given that the winner-take-all mechanism uses nonlinear dynamics. Lastly, we found that the EI-NODE model outperforms the EI-RNN baseline, most likely due to the increased expressivity allowed in the hidden layers of the NODE models compared to the fixed recurrent weight of the RNN (Sedler et al., 2023).

3.2 Results: Dynamics Interpretability

The EI-NODE model recovered the dynamics of the winner-take-all system, including the two distinct fixed-point attractors that correspond to the circuit's stable decision states (Fig. 3A). Critically, the model's interpretable structure allowed us to decompose the learned dynamics into the influences from each population at a given time (Fig. 3, Appendix A.1). This analysis is not achievable with standard NODE models (as dynamics influences are not cleanly partitioned across populations), and influences extracted from RNNs are not accurately represented or as easily interpretable because the sum of influences is wrapped in a nonlinearity. Extracting the influences over time for the EI-NODE model confirmed that the stable states emerged from a balance of strong self-excitation within the winning population and targeted inhibition from the shared I pool, where individual population influence arrows cancel out at fixed points (Fig. 3A bottom, colored arrows; Fig. 3B at the end of the trial). The early increase in activity is primarily driven by self-excitation, prior to the I population increasing enough to balance out this excitation (Fig. 3A top, colored arrows; Fig. 3B early in the trial). This fixed point structure arises early in trials (around 1400ms) for nonlinear models, but is not found by the linear models (Fig. 3C, D) or RNN (See Appendix, Figure 3). Additionally, the influences of different populations on dynamics learned by the linear model are not as easily interpretable with respect to the true underlying biological system. For instance, inhibitory latents

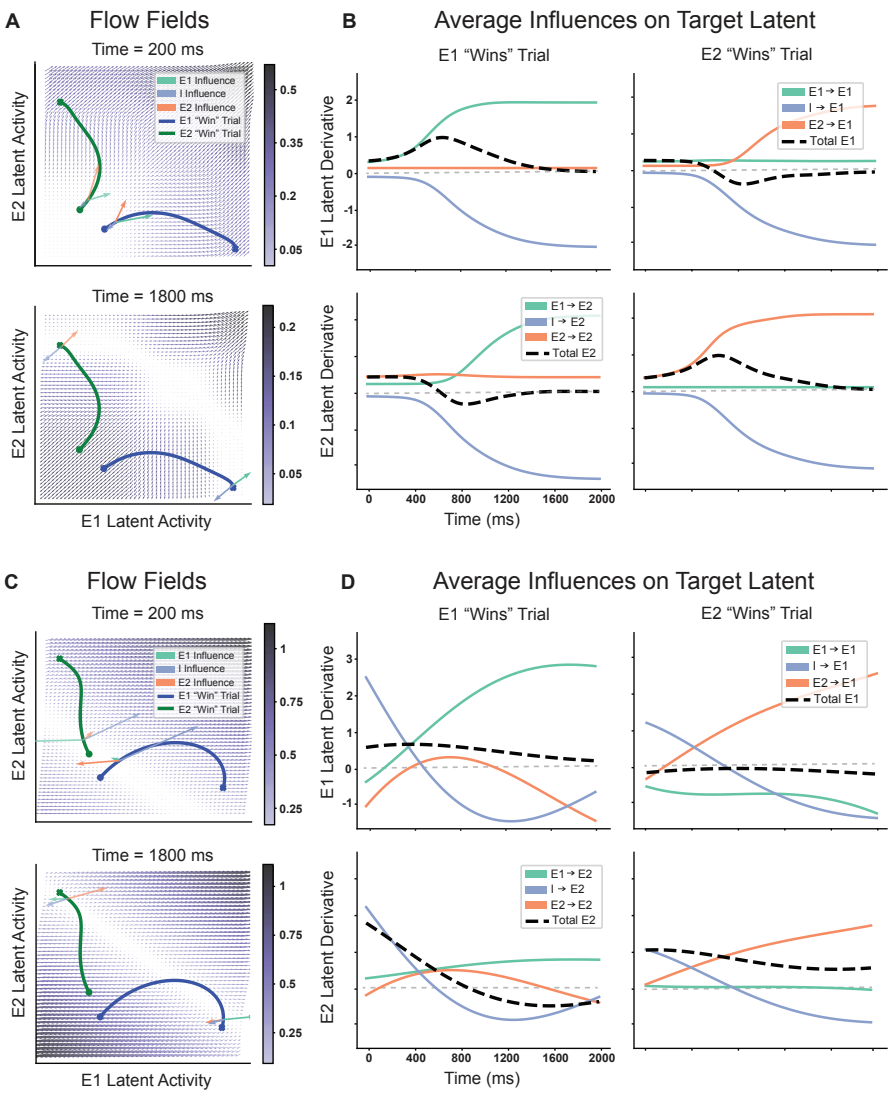

Figure 3: *Synthetic data dynamics interpretability*. **A)** EI-NODE latent vector fields (E1 vs. E2) at the trial-averaged I activity early (200 ms, top) and late (1800 ms, bottom) in the trial, overlaid with trial-averaged trajectories for E1 'win' (blue) and E2 'win' (green) trials. Colored arrows show each population's contribution to the vector field: strong self-excitation from the winning population early, and cancellation between the winning E and I populations late, indicating a fixed point. **B)** Trial-averaged influence of each population on E1 (top) and E2 (bottom), by trial type. **C, D)** Same as **A, B** for the Linear EI-NODE, which shows less biologically realistic early dynamics and no fixed point. Flow fields for other models: Appendix E.2.5, Figs. 4 and 5.

can appear to cause a positive change in dynamics (because when the latent has a negative value, multiplying by its negative weight will lead to a positive output). Overall, these results serve as a successful proof-of-concept, validating that our framework can accurately learn and expose the underlying mechanisms of a nonlinear neural dynamical system of E and I populations.

3.3 Results: Learning Unlabeled Cell Types

In actual experiments, cell types of each recorded neuron are not always known. We thus tested the ability of our model to infer the cell type labels when only a fraction of the neurons' cell types were known. We modified the readout mask (M_{pop} in equation 5) to have a block-diagonal structure with only two blocks, one for both E populations (E1 and E2) and one for the I population. This is because an experimentalist may only know E versus I, rather than knowing E1 or E2 *a priori*, as

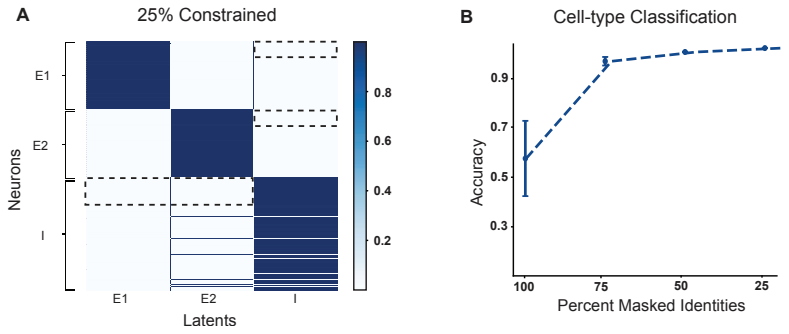

Figure 4: *Learning unlabeled cell types*. **A**) EI-NODE (normalized) readout weights for a model initialized with 75% of cell-type identities masked. Dashed boxes: readout weights constrained to 0, so that 25% of excitatory neurons cannot map to the inhibitory latent and 25% of inhibitory neurons cannot map to the excitatory latents; all other weights were learned. **B**) Same as **A** across varying levels of cell-type identity masking.

we previously assumed. We then unmasked the off-diagonal sections for a subset of neurons (i.e. masked their identities) before training the EI-NODE model with Hoyer-Square regularization on the unmasked readout weights to encourage sparsity (outlined in Section A.4). Using this approach, we find that the EI-NODE model can accurately infer the cell-type identities of most of the unlabeled neurons. We tested the EI-NODE model's ability to recover cell-type identities at four different levels of identity masking: 100%, 75%, 50%, and 25% (Figure 4). At 75% masked, the EI-NODE model is able to recover cell-type identities with 94.1% ($\pm$3.3%) accuracy (Fig. 4A). Each row's (neuron's) weights are normalized to have a maximum value of 1. The color of each cell in a row denotes the relative amount each latent is used to predict a given neuron's activity, therefore the cell-type predictions were taken to be the argmax of each row in the normalized readout weights. When masking half and a quarter of the neuron identities, the model can identify the cell-types of these unidentified neurons with an accuracy of 97.3% ($\pm$ 0.3%) and 98.6% ($\pm$ 0.1%), respectively. Even when training on a completely unlabeled set of neurons (100% masking), the model can achieve an accuracy of 60.9% ($\pm$20.6%). This provides a proof of principle for the ability to apply EI-NODE models when not all cell classes are known, and additionally, to discover these unknown cell class identities.

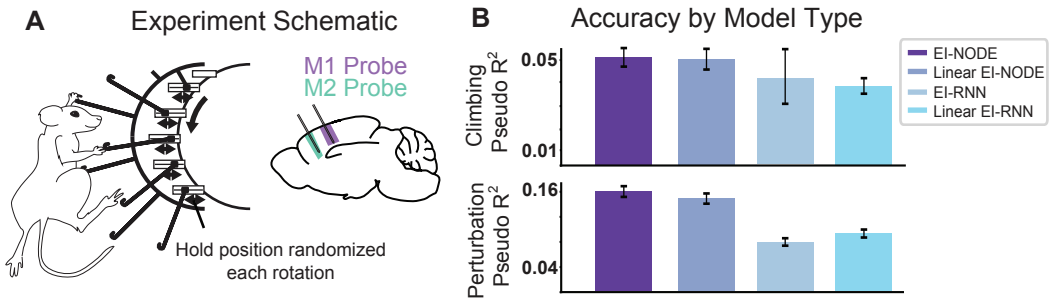

Figure 5: *Naturalistic Climbing and Optogenetics Setup, Model Performance.* **A)** Schematic describing mouse task. Probes inserted into M1 and M2 recorded neural activity during self-paced climbing. On a subset of trials inhibitory neurons within M1 were activated optogenetically. **B)** Future prediction accuracy, quantified by median pseudo-R^2 across all neurons, is compared across model types, both during climbing without optogenetic perturbations (top) and following optogenetic perturbations (bottom).

4 Application to Neural Recordings during Naturalistic Climbing with Optogenetics

Another capability of our method is modeling optogenetic perturbations, which typically target a specific cell type. Thus, we applied EI-NODE to neural data recorded in mice undergoing optogenetic perturbations while they performed a naturalistic, self-initiated all-limb climbing task (Kristl et al., 2025) (Fig. 5A). Mice were placed in a climbing apparatus where the contralateral handholds were randomly positioned each time, so as to require consistent, yet flexible behavior. Neuropixels probes were used to simultaneously record from the premotor (M2) and primary motor (M1) equivalent regions in rodents, the rostral and caudal forelimb areas, respectively. On a subset of trials during climbing, optogenetic activation of inhibitory neurons in either M1 or M2 (depending on the animal) occurred for brief 25 ms stimulation windows. To provide the distinct sub-population inputs required by EI-NODE, we first needed to classify the recorded units by cell class. Targeted inhibitory neurons were identified using a strict response window for opto-tagging, while remaining units were classified via validated waveform-width criteria (e.g., McCormick et al., 1985; Barthó et al., 2004, See Appendix B.2 for more details). We fit our model on two sessions in which M1 and M2 were independently optogenetically targeted. The resulting dataset contained four populations of neurons with neuron counts of: M1-E (38 and 48 neurons), M1-I (26, 22), M2-E (109, 82), and M2-I (16, 44). Models were fit to data that incorporated both timepoints at which perturbations occurred and timepoints during natural climbing.

4.1 Model Fitting

For this dataset, we customized our model fitting in a few ways. First, inputs were provided to the model at the time of the optogenetic perturbations, specifically to the targeted population (details in Appendix B.2.1). Second, to further biologically constrain across-region interactions, we utilized connectivity-constrained models for these experiments. Specifically, we restrict inhibitory connections to be within-region only, consistent with anatomical measurements. Lastly, to create a more comprehensive model that incorporated optogenetics to both regions, we fit a joint model to both sessions with a shared underlying dynamics network and individual session-specific encoders and decoders.

4.2 Results

As in the synthetic dataset, we compared our model's accuracy with several other architectures (Fig. 5B, Table 5)), but only for models that allow cell-class-specific perturbations (for instance, a standard NODE model cannot model the optogenetic activation of inhibitory neurons). Nonlinear EI-NODE very slightly outperformed a linear version, and substantially improved over EI-RNNs, again demonstrating how this architecture can improve performance while maintaining biological interpretability.

Indeed, the EI-NODE model successfully learns to model average neural activity during optogenetic activation across all neural populations (both regions and cell-classes; Fig. 6A). The targeted population (M1 I in this figure) responds accurately to the modeled optogenetic input, as shown by a steep jump in its activity in the first time bin in Fig. 6A. The corresponding excitatory population (M1 E) activity dips, then slowly rebounds. Importantly, our model formulation clearly shows how these populations influence one another over this time period to produce this activity. We can see this perturbation response indeed results from a phasic increase in negative drive coming from M1 I, which slowly gives way to self-excitation (Fig. 6B).

We then further used our framework to inspect inter-population influences and develop biological hypotheses about the dynamics during climbing across sessions. By analyzing the learned influences found within our EI-NODE models, we found that M1 E strongly excites its own local and cross-region connections, whereas M2 E shows minimal cross-region and weaker within-region influences (Fig.6C). This supports recent experimental findings that M1 dominates over M2 during climbing, unlike during reaching (Kristl et al., 2025), and demonstrates how EI-NODE can be used to develop cell-class specific hypotheses about inter-population dynamics.

Lastly, to create greater confidence in EI-NODE's scientific results above, we created a simulated RNN-based dataset with the 4 populations, both during climbing and optogenetic perturbations (Ap-

pendix E.4). EI-NODE was indeed able to accurately capture the relative interactions between populations (Fig.6; in fact, more so than an EI-RNN Fig.7), further demonstrating the potential of EI-NODE models.

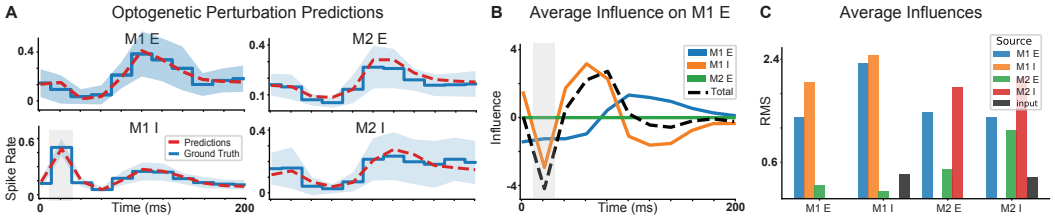

Figure 6: *Model Interpretability during Optogenetic Perturbations.* **A)** Average population activity during optogenetic-activation of M1 I at timebin 1. Binned ground-truth activity shown in blue, with standard deviation in light blue, and EI-NODE predictions shown in dashed red. **B)** Learned influence on coupled M1 E population to the targeted M1 I optogenetic target. Activity sharply decreases at the time of perturbation (shown as light gray overlay), driven by the large, negative M1 I influence, shown in orange. **C** Learned EI-NODE population influences for each source-target pair. Bar height is the root-mean-square (RMS) of that influence over validation trials, time, and target latent units, averaged with equal weight across sessions and across climbing and perturbation. EI-NODE finds a strong and unbalanced local and inter-region M1 E drive, whereas M2 E displays weaker cross- and within-region influences, consistent with recent findings (Kristl et al., 2025).

5 Discussion

Here, we introduced Excitatory-Inhibitory NODE (EI-NODE) models that enable modeling the interactions between E and I populations within a flexible nonlinear dynamical system. In synthetic and recorded Neuropixels probe datasets, we demonstrated how they retain the prediction accuracy of standard NODE models, while providing a newfound interpretability of how constituent populations drive the overall dynamics of a system.

A model limitation is assuming that we know the cell type identities of all neurons. Classifying neurons as E or I based on waveform characteristics, as we did with the recorded neural data, is an imperfect heuristic. While we showed initial results that we could learn cell types with partial knowledge—in particular, if a portion of the neurons had known cell classes, it would be valuable for future work to extend this line of inquiry. For instance, to model the scenario where we are unsure about many cell types (e.g. due to imperfect waveform classification), we could couple L1 or Hoyer regularization for sparsity, with Tikhonov regularization to penalize readout weights inversely to the confidence we have about the classification.

Another central limitation of our current work is that the NODEs are fully deterministic in their dynamics and do not model noise in the underlying dynamics. Thus, the way that the model accounts for single-trial noise and variability is fully through the learned initial condition. The 'ground-truth' in the synthetic data is that the initial conditions for trials in which the two different E populations end up winning are actually almost identical—rather, a combination of input noise early on, coupled with the E/I dynamics, drives the model towards one solution. Using deterministic NODEs to best model this system, the models rather learned two separate locations of initial conditions for the trials in which the two E populations won. Thus, it would be valuable in future work to extend our EI-NODE framework to neural stochastic differential equations (NSDEs), which have successfully modeled neural activity (Kim et al., 2023). The dynamics architecture we developed should be readily interchangeable for standard NODE dynamics within NSDEs. In fact, our proposed dynamics architecture could more generally offer a path towards understanding cell-type-constrained interactions within a broad range of architectures of nonlinear dynamical systems modeling (Hernandez et al., 2018; Karniol-Tambour et al., 2022; Liu et al., 2025).

AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on implemented methods, cleaning and reformatting datasets, and supporting qualitative data analysis. We have not used generative AI tools for proposing or refining hypotheses, generating synthetic datasets, helping develop theoretical models or conceptual frameworks, or interpreting results. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work.

Additionally, we used generative AI tools for creating some figures within the supplementary material, editing software for readability, and checking general formatting and grammar within the manuscript. We have reviewed all AI-assisted work and manually checked any generated code for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI, or assist with translation.

ETHICS STATEMENT

Regarding the climbing dataset, all experiments and procedures involving animals were performed in accordance with NIH guidelines and approved by IACUC.

REPRODUCIBILITY STATEMENT

Anonymized code to view the model implementation and training pipeline is available here: `https://anonymous.4open.science/r/node-5E24/`. The synthetic simulated dataset is open-sourced and was generated using the default circuit parameters in the provided script. Details for generating the synthetic dataset and data preprocessing and labeling for the naturalistic climbing dataset are discussed in Appendix B. We will make the climbing dataset used to train our models publicly available upon acceptance.

REFERENCES

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

Jeffrey N Stirman, Ikuko T Smith, Michael W Kudenov, and Spencer L Smith. Wide field-of-view, multi-region, two-photon imaging of neuronal activity in the mammalian brain. *Nature biotechnology*, 34(8):857–862, 2016.

Misha B Ahrens, Michael B Orger, Drew N Robson, Jennifer M Li, and Philipp J Keller. Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nature methods*, 10 (5):413–420, 2013.

Britton A Sauerbrei, Jian-Zhong Guo, Jeremy D Cohen, Matteo Mischiati, Wendy Guo, Mayank Kabra, Nakul Verma, Brett Mensh, Kristin Branson, and Adam W Hantman. Cortical pattern generation during dexterous movement is input-driven. *Nature*, 577(7790):386–391, 2020.

Takahide Itokazu, Masashi Hasegawa, Rui Kimura, Hironobu Osaki, Urban-Raphael Albrecht, Kazuhiro Sohya, Shubhodeep Chakrabarti, Hideaki Itoh, Tetsufumi Ito, Tatsuo K Sato, et al. Streamlined sensory motor communication through cortical reciprocal connectivity in a visually guided eye movement task. *Nature communications*, 9(1):1–14, 2018.

Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. *Advances in neural information processing systems*, 24, 2011.

Biljana Petreska, Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Dynamical segmentation of single trials from population neural data. *Advances in neural information processing systems*, 24, 2011.

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

Timothy D Kim, Thomas Z Luo, Jonathan W Pillow, and Carlos D Brody. Inferring latent dynamics underlying neural population activity via neural differential equations. In *International Conference on Machine Learning*, pages 5551–5561. PMLR, 2021.

Andrew R Sedler, Christopher Versteeg, and Chethan Pandarinath. Expressive architectures enhance interpretability of dynamics-based neural population models. *Neurons, behavior, data analysis, and theory*, 2023:10–51628, 2023.

Thomas Zhihao Luo, Timothy Doyeon Kim, Diksha Gupta, Adrian G Bondy, Charles D Kopec, Verity A Elliott, Brian DePasquale, and Carlos D Brody. Transitions in dynamical regime and neural mode during perceptual decisions. *Nature*, pages 1–11, 2025.

Stephen L Keeley, David M Zoltowski, Mikio C Aoi, and Jonathan W Pillow. Modeling statistical dependencies in multi-region spike train data. *Current opinion in neurobiology*, 65:194–202, 2020.

Matthew G Perich, Charlotte Arlt, Sofia Soares, Megan E Young, Clayton P Mosher, Juri Minxha, Eugene Carter, Ueli Rutishauser, Peter H Rudebeck, Christopher D Harvey, et al. Inferring brain-wide interactions using data-constrained recurrent neural network models. *BioRxiv*, pages 2020–12, 2020.

Belle Liu, Jacob Sacks, and Matthew D Golub. Accurate identification of communication between multiple interacting neural populations. *arXiv preprint arXiv:2506.19094*, 2025.

Matthew Dowling and Cristina Savin. Nonlinear multiregion neural dynamics with parametric impulse response communication channels. In *The Thirteenth International Conference on Learning Representations*.

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

Weihan Li, Chengrui Li, Yule Wang, and Anqi Wu. Multi-region markovian gaussian process: An efficient method to discover directional communications across multiple brain regions. *Proceedings of machine learning research*, 235:28112, 2024.

Sai Koukuntla, Joshua B Julian, Jesse C Kaminsky, Manuel Schottdorf, David W Tank, Carlos D Brody, and Adam S Charles. Unsupervised discovery of the shared and private geometry in multi-view data. *arXiv preprint arXiv:2408.12091*, 2024.

Trisha Jha, Omid G Sani, Bijan Pesaran, and Maryam M Shanechi. Prioritized learning of cross-population neural dynamics. *Journal of Neural Engineering*, 2025.

Evren Gokcen, Anna I Jasper, João D Semedo, Amin Zandvakili, Adam Kohn, Christian K Machens, and Byron M Yu. Disentangling the flow of signals between populations of neurons. *Nature Computational Science*, 2(8):512–525, 2022.

Evren Gokcen, Anna Jasper, Alison Xu, Adam Kohn, Christian K Machens, and Byron M Yu. Uncovering motifs of concurrent signaling across multiple neuronal populations. *Advances in Neural Information Processing Systems*, 36:34711–34722, 2023.

Joao D Semedo, Amin Zandvakili, Adam Kohn, Christian K Machens, and Byron M Yu. Extracting latent structure from multiple interacting neural populations. *Advances in neural information processing systems*, 27, 2014.

Joshua Glaser, Matthew Whiteway, John P Cunningham, Liam Paninski, and Scott Linderman. Recurrent switching dynamical systems models for multiple interacting neural populations. *Advances in neural information processing systems*, 33:14867–14878, 2020.

Orren Karniol-Tambour, David M Zoltowski, E Mika Diamanti, Lucas Pinto, David W Tank, Carlos D Brody, and Jonathan W Pillow. Modeling communication and switching nonlinear dynamics in multi-region neural activity. *bioRxiv*, pages 2022–09, 2022.

H Francis Song, Guangyu R Yang, and Xiao-Jing Wang. Training excitatory-inhibitory recurrent neural networks for cognitive tasks: a simple and flexible framework. *PLoS computational biology*, 12(2):e1004792, 2016.

Aditi Jha, Diksha Gupta, Carlos Brody, and Jonathan W Pillow. Disentangling the roles of distinct cell classes with cell-type dynamical systems. *Advances in Neural Information Processing Systems*, 37:33668–33690, 2024.

Astrid A Prinz, Dirk Bucher, and Eve Marder. Similar network activity from disparate circuit parameters. *Nature neuroscience*, 7(12):1345–1352, 2004.

Abhranil Das and Ila R Fiete. Systematic errors in connectivity inferred from activity in strongly recurrent networks. *Nature Neuroscience*, 23(10):1286–1296, 2020.

Manuel Beiran and Ashok Litwin-Kumar. Prediction of neural activity in connectome-constrained recurrent networks. *bioRxiv*, 2024.

Hennie Daniels and Marina Velikova. Monotone and partially monotone neural networks. *IEEE Transactions on Neural Networks*, 21(6):906–917, 2010.

Davide Sartor, Alberto Sinigaglia, and Gian Antonio Susto. Advancing constrained monotonic neural networks: Achieving universal approximation beyond bounded activations. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

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

Timothy Doyeon Kim, Thomas Zhihao Luo, Tankut Can, Kamesh Krishnamurthy, Jonathan W Pillow, and Carlos D Brody. Flow-field inference from neural data using deep recurrent networks. *bioRxiv*, 2023.

Xiao-Jing Wang. Probabilistic decision making by slow reverberation in cortical circuits. *Neuron*, 36(5):955–968, 2002.

Marcel Stimberg, Romain Brette, and Dan F. M. Goodman. Brian 2, an intuitive and efficient neural simulator. *eLife*, 8:e47314, 2019. doi: 10.7554/eLife.47314. URL https://doi.org/10.7554/eLife.47314.

Amy Kristl, Natalie Koh, Mark Agrios, Sajishnu Savya, Zhengyu Ma, Diya Basrai, Sarah Hsu, and Andrew Miri. Interactions between motor cortical forelimb regions and their influence on muscles reorganize across behaviors. *bioRxiv*, pages 2025–01, 2025.

David A. McCormick, Barry W. Connors, James W. Lighthall, and David A. Prince. Comparative electrophysiology of pyramidal and sparsely spiny stellate neurons of the neocortex. *Journal of Neurophysiology*, 54(4):782–806, 1985.

Peter Barthó, Hajime Hirase, Lénaïc Monconduit, Michael Zugaro, Kenneth D. Harris, and György Buzsáki. Characterization of neocortical principal cells and interneurons by network interactions and extracellular features. *Journal of Neurophysiology*, 92(1):600–608, 2004.

Daniel Hernandez, Antonio Khalil Moretti, Ziqiang Wei, Shreya Saxena, John Cunningham, and Liam Paninski. Nonlinear evolution via spatially-dependent linear dynamics for electrophysiology and calcium data. *arXiv preprint arXiv:1811.02459*, 2018.

Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004. URL https://www.jmlr.org/papers/volume5/hoyer04a/hoyer04a.pdf.

Huanrui Yang, Wei Wen, and Hai Li. Deephoyer: Learning sparser neural network with differentiable scale-invariant sparsity measures. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rylBK34FDS.

Matthew G. Perich, Charlotte Arlt, Sofia Soares, Siyan Zhou, Manuel Beiran, Aaron S. Andalman, Tyler Benster, Megan E. Young, Clayton P. Mosher, Juri Minxha, Eugene Carter, Ueli Rutishauser, Peter H. Rudebeck, Christopher D. Harvey, Karl Deisseroth, and Kanaka Rajan. Inferring brain-wide interactions using data-constrained recurrent neural network models. *Neuron*, 2026.

A Extended Methods

A.1 Extracting Influences from EI-NODE models

Defining the latent dynamics for the EI-NODE model as the sum of interactions between/within populations allows us to extract the influence each population k has on the total derivative for population j. This is best illustrated with an example using the populations from Section 3. Expanding equation 4 for the latent dynamics of population E1:

$$\frac{dz^{E1}}{dt} = f_\theta^{E1,E1}(z^{E1}) + \tilde{f}_\theta^{E1,E1}(z^{E1}) + \tilde{f}_\theta^{E1,E2}(z^{E2}) + \tilde{f}_\theta^{E1,I}(z^{I})$$

where f_θ is an unconstrained linear layer that acts independently on each latent for population E1 and each $\tilde{f}_\theta$ is a constrained monotonic MLP. Depending on the identity of population k, the output of $\tilde{f}_\theta^{jk}$ is constrained to be positive (negative) if the population is excitatory (inhibitory). Therefore, each term gives us the influence of each population onto population E1 over time (Figure 3B top panels). In the influence plots, the learned self influence (in this example $f_\theta^{E1,E1}(z^{E1})$) is added to the influence of the constrained interaction (in this example $\tilde{f}_\theta^{E1,E1}(z^{E1})$).

A.2 Ensuring Monotonicity using the Post-Activation Switch

To ensure the learned population interactions are monotonic, we parameterized the networks f_θ^{jk} using post-activation switch networks introduced by Sartor et al. (2025). By reordering the computational steps of a feedforward layer, the authors show that any four-layer MLP (with convex monotone activations) can approximate a monotonic function without needing to constrain the weights to be nonnegative. This alleviates known difficulties in optimizing nonnegative parameterized networks, such as sensitivity to initialization and vanishing/exploding gradients.

Let $f(x)$ be a single-layer transformation $f(x) = \sigma(|W|x + b)$ where the absolute weights $|W|$ enforce non-negativity, then the post-activation switch defines an equivalent expression of $f(x)$ as:

$$\hat{f}(x) = W^+\sigma(x) + W^-\sigma(-x) + b, \tag{7}$$

where x is the input to the layer, b is the bias term, W^+ are the positive components of the weights ($\max(W, 0)$), W^- are the negative components of the weights ($\min(W, 0)$), and σ is any convex, monotonic activation function. In this work, for both datasets, f_θ^{jk} in equation 4 are four-layer MLPs with $\sigma(x) = \tanh(x)$.

A.3 Nonnegative Weight Parameterization

The nonnegative constraint on linear network weights cannot be relaxed using the method introduced by Sartor et al. (2025). Therefore, the weights, W, of linear monotonic networks or modules in this work were parameterized to be nonnegative or nonpositive using the $\mathrm{softplus}$ activation

$$W \leftarrow \pm\,\mathrm{softplus}(W)$$

Nonpositive weight parameterization was used for the excitatory dynamics functions, f_θ^{jk}, of the Linear EI-NODE model and the block diagonal readouts across all models. Nonnegative weight parameterization was only used for the inhibitory dynamics functions of the Linear EI-NODE model. The $\mathrm{softplus}$ activation was chosen over ReLU or $\exp$ since its derivative is defined and bounded, preventing the gradient from vanishing/exploding.

A.4 Hoyer Regularization for Cell-type Identification Experiments

For the model to identify cell-types for each neuron, it must find sparse readout solutions with non-zero weights only for latents of the identified population. Therefore, we utilized Hoyer Square regularization (Hoyer, 2004) (Yang et al., 2020) on the semi-constrained readout matrix, $W_{\text{dec}} \in$

$\mathbb{R}^{NxL}$, to encourage sparsity along the latent dimension (N is the number of neurons and L is the number of latents):

$$\mathrm{H}_S(W_{\text{dec}}) = \sum_{n=1}^{N} \frac{\left(\sum_{l=1}^{L} |w_{n,l}|\right)^2}{\sum_{l=1}^{L} w_{n,l}^2} \tag{8}$$

where $w_{n,l}$ is the weight in W_{dec} for neuron n and latent l. We added the sparsity term to the loss function with a penalty λ_H:

$$\mathcal{L}_\theta(X, \hat{Y}) = -\frac{1}{TN} \sum_{t=1}^{T} \sum_{n=1}^{N} \log \text{Poisson}\left(s_{t,n} \mid \hat{x}_{t,n}\right) + \lambda_H * \mathrm{H}_S(W_{\text{dec}})$$

For the cell-type recovery experiments in Section 3.3, λ_H was set to 0.00001. In the case where there are multiple latents per population, the Group Hoyer Regularization can be applied to encourage a group of latents to be non-zero:

$$G_{\mathrm{H}_S}(W_{\text{dec}}) = \sum_{n=1}^{N} \frac{\left(\sum_{g=1}^{G} ||w_{n,g}||_2\right)^2}{||W_{\text{dec}}||_2^2} \tag{9}$$

This applies the Hoyer-Square regularization over G groups instead of individually to each latent l, where $w_{n,g}$ is a vector of weights corresponding to neuron n and a group of latents, g, belonging to a single neural population. Importantly, this definition of $G_{\mathrm{H}_S}(W_{\text{dec}})$ only holds if the set of all groups ($g \in G$) are non-overlapping.

B Training and Dataset Details

As discussed in Section 2.4, all models were trained using SGD and an AdamW optimizer. Models were trained using a curriculum learning protocol where the sequence length of all trials were shortened at the start of training to length $\tilde{T}_{\text{start}}$ and linearly increased to the full sequence length T from epoch 0 to a prespecified amount of warmup epochs. During any epoch in the warmup period, the segment of time for each sample within a batch (with length $\tilde{T}_{\text{start}} \leq T_{\text{warmup}} < T$) was randomly sampled from the trial. Other important details are outlined for each dataset below.

Model hyperparameters were found using sweeps to find the combination of parameters that minimized the validation loss. For each dataset, the results reported in the paper were cross-validated across 5 training/validation folds. Models were trained on Apple Silicon Macbook Pros and Slurm CPU clusters. On the simulated and climbing datasets, training EI-NODE models took about 4 hours and 1.5 hours, respectively. Training the other baseline models in C on the simulated and climbing datasets took about 2 hours and 0.5 hours, respectively.

B.1 Simulated Dataset

The Brian2 package (Stimberg et al., 2019) was used to generate spiking activity from populations of neurons (two excitatory, one inhibitory) performing a simulated noisy sensory integration decision-making task described in (Wang, 2002). The general connectivity structure of the populations are as follows: within each excitatory pool, the recurrent weights are strong ($|w| > 1$) and mediated by slow NMDA receptors; across excitatory pools, the weight strengths are weaker ($|w| < 1$); and the weight strengths from the inhibitory pool to all excitatory pools are of equal strength ($|w| = 1$) to provide broad, global inhibition. Given this connectivity structure, the populations follow a consistent dynamical rule: Stimuli to each population drives a self-excitation loop into E1 and E2 (facilitating integration). Elevated activity within the E populations drives fast inhibitory activity (GABA-A mediated) back onto themselves. The population with the weaker activity level (dependent on the strengths of input stimuli) becomes suppressed by the global inhibitory signal resulting

in the winner-take-all dynamic. The winning population activity becomes stabilized by the global inhibition, and any excitation by the winning E population onto the losing E population is silenced by the global inhibition.

Trials were generated at low and high stimulus coherence levels for both trial conditions ("E1 win" and "E2 win" trials, 4 conditions total). The stimulus coherence level was set to 6.4 and 25.6 for hard and easy trial conditions, respectively. Models were trained to reconstruct spiking activity from two-second sequences where either population E1 or E2 would exhibit a sustained increase in spiking activity (Winner-Take All; examples in Fig. 2A). A total of 320 trials were generated and randomly split into a training and validation set, with 80% of the trials used for training and 20% for validation. For each trial, the spiking activity was binned into 20 ms bins, creating a trial sequence length of 100 bins. Each 100-bin trial was input into each model to obtain the initial latent condition before rolling out the dynamics and aiming to reconstruct the original sequence of neural activity. In section 3, the latents were partitioned by population with one latent per population (two E latents and one I latent). All models were trained for 2000 epochs using curriculum learning for 1500 warmup epochs ($\tilde{T}_{\text{start}} = 20$), a batch size of 32, and a learning rate of 0.001.

B.2 Naturalistic Climbing Dataset

Models were trained using the binned spiking data of 189 neurons recorded simultaneously from M1 and M2 in a single mouse. Consistent with prior work (Kristl et al., 2025), neurons were initially segregated into putative excitatory (E) and inhibitory (I) classes using waveform analysis. We designated neurons as opto-tagged if their activity exceeded baseline by more than 0.35 standard deviations during the 15 ms window immediately following laser stimulation. For spike sorting, we utilized Kilosort3 with standard default settings. We excluded any units exhibiting $> 20\%$ refractory period violations in their autocorrelograms. The remaining Kilosort "good units" underwent manual examination to verify the biological plausibility of their autocorrelograms, waveform consistency, and spike amplitudes. To guard against light-induced electrical artifacts, we aligned spike times to the laser pulse and removed a small subset of units located in the optical path. Light-induced artifact spikes were easily flagged because they occurred exactly at laser onset or offset across all trials, without the natural, sub-millisecond temporal fluctuation inherent to true light-activated interneurons. We reclassified 26 M1-E neurons to M1-I and 29 M2-E neurons to the M2-I pool that were highly modulated by optogenetic activation (separate sessions).

In total, our dataset contained four populations of neurons across two sessions: M1-E (38 and 48 neurons), M1-I (26, 22), M2-E (109, 82), and M2-I (16, 44). All spiking activity was binned into 20 ms bins and neurons with a firing rate less than 0.5 Hz across all trials were labeled as "silent" and removed. Given the extreme heterogeneity of climbing periods in this self-initiated task, trials were created by aligning to peak activity within a window of 200ms after rotary encoder movement detection. For optogenetic activation trials, activity was taken 20ms before to 120 ms after the 25 ms laser pulse.

Models were trained and evaluated on a total of 227 trials when the mouse was actively climbing and 88 optogenetic perturbation trials. We used a random 80% of the trials for training and 20% for validation, for which all reported values are reported. Population latent dimensionalities used for the dynamics models were sized to reflect the relative number of neurons in each neural population (10 for E and 5 for I populations, resulting in 30 latents total). All models were trained for 3000 epochs using curriculum learning for 1500 warmup epochs ($\tilde{T}_{\text{start}} = 20$), a batch size of 32, and a learning rate of 0.001.

B.2.1 Incorporating Inputs into EI-NODE Models to Model Optogenetic Data

The model received sequences of 20-ms binned spike counts from the M1-E, M1-I, M2-E, and M2-I populations, concatenated with four time-dependent input channels (one per population). For climbing trials, all input channels were zero. For perturbation trials, the input was represented as a unit pulse in the optogenetically targeted inhibitory population at the second time bin: M1 I or M2 I depending on the session. All other input values were zero. The spike and input sequences were provided jointly to the session-specific encoder to estimate the initial latent state. The same input sequence was passed to the latent dynamics at each time step, where a learned, bias-free

nonlinear mapping transformed the targeted pulse into an additive contribution to the corresponding population's latent derivative.

C MODELS

As stated in Section 2.4, multiple dynamics models were trained to serve as comparison points to the EI-NODE model. Across all dynamics models, a one-layer bidirectional GRU (BiGRU) combined with a linear network maps the input spike trains to an initial latent z_0 outlined in 3. To counteract any overfitting, a dropout layer with probability $p = 0.05$ was placed after the BiGRU encoder and before the linear mapping to the latents during training. In standard RNN notation, the GRU cell is defined by the following equations:

$$\begin{aligned} r_t &= \sigma(W_{ir}s_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}) \\ z_t &= \sigma(W_{iz}s_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}) \\ n_t &= \tanh(W_{in}s_t + b_{in} + r_t * (W_{hn}h_{t-1} + b_{hn})) \\ h_t &= z_t * h_{t-1} + (1 - z_t) * n_t \end{aligned} \tag{10}$$

where s_t is the input (binned spike counts at time t), h_t is the current hidden state, h_{t-1} is the previous hidden state, r_t is the reset gate (controlling how much of the past information to forget), z_t is the update gate (controlling how much new information to add or past information to retain), n_t is the candidate hidden state, and σ is the sigmoid function. The hidden state h_t was set to 80 per direction (160 combined).

In addition to the shared encoder definition, the decoding module defined by equation 5 was held constant across baselines. Therefore, the only varied model component was the dynamics model $f_\theta(z)$ in equation 1. Below, the details of each baseline dynamics model are outlined.

C.1 EI-NODE

EI-NODE is a variant of NODE model where the dynamics are explicitly divided into population dynamics functions that are constrained to be nonnegative for E populations and nonpositive for I populations. The dynamics of this model are described by equation 4 For every target population and every source population, including the target itself, a three-layer monotonic MLP with hidden widths 64 and 64maps that source block onto the target block. Each layer has a bias, and $\tanh$ is applied to the hidden layers. Population latent sizes were assigned based on the size of the recorded populations, 10 for excitatory and 5 for inhibitory populations.

C.2 LINEAR EI-NODE

As the name implies, the Linear EI-NODE is the linear version of the EI-NODE model described in Section 2.3. The critical distinction is that f_θ^{jk}, do not contain a nonlinear activation function, and their outputs are not wrapped in a softplus activation to avoid nonlinear effects.

C.3 UNCONSTRAINED NONLINEAR AND LINEAR NODES

The dynamics for the Unconstrained Nonlinear and Linear NODEs are parameterized by a vanilla multilayer perceptron (MLP) with N layers:

$$\begin{aligned} dz/dt &= W_N a_{N-1} + b_N, \\ a_{N-1} &= \phi(W_{N-1}a_{N-1} + b_{N-1}) \\ &\vdots \\ a_2 &= \phi(W_2 a_1 + b_2) \\ a_1 &= \phi(W_1 z_{t-1} + b_1) \end{aligned} \tag{11}$$

The activation function, ϕ, is the $\tanh$ activation function for the Nonlinear NODE and the Identity function for the Linear NODE. Weight matrices W_l for $l = 0, \ldots, N$ are not constrained to be monotonic. The number of parameters in the NODE models was matched to the number of parameters in the EI-NODE models for fair comparison.

C.4 VANILLA AND DALE'S LAW CONSTRAINED RNN

The dynamics for both the Vanilla and Dale's Law constrained RNNs are defined by the standard RNN equation with no inputs:

$$dz/dt \;=\; \tanh(W_{\mathrm{rnn}} z_{t-1} + b_{\mathrm{rnn}}) \tag{12}$$

For the vanilla RNN, the weight matrix W_{rnn} is not constrained to be nonnegative. For the Dale's Law constrained RNN, the columns of W_{rnn} associated with inhibitory latents are negative and the columns associated with excitatory latents are positive. Additionally, the diagonal elements are unconstrained, allowing the latents to have either a positive or negative influence on itself. These sign constraints are implemented using softplus weight parameterizations discussed in A.3. Importantly, just as in the Linear EI-NODE model, latent activations are not constrained to be positive/negative using nonlinearities. Therefore, an inhibitory (excitatory) latent may have an excitatory (inhibitory) effect if its activation is negative. This hurts the interpretability of these models when applied to dynamical systems frameworks.

D METRICS

D.1 PSTH R^2 SCORE

The peri-stimulus time histogram (PSTH) R^2 measures the coefficient of determination between the PSTHs of the true spiking activity and the model's predicted firing rates. Before calculation, the spiking activity for each trial is smoothed using a Gaussian kernel to get estimated firing rates. Let $x_{k,t,n}$ denote the smoothed spike count and $\hat{x}_{k,t,n}$ the model-predicted firing rate on trial $k \in \{1,\dots,K\}$, time bin $t \in \{1,\dots,T\}$, and neuron $n \in \{1,\dots,N\}$. The PSTH for each neuron is obtained by averaging across trials,

$$\bar{x}_{t,n} \;=\; \frac{1}{K}\sum_{k=1}^{K} x_{k,t,n}, \qquad \hat{\bar{x}}_{t,n} \;=\; \frac{1}{K}\sum_{k=1}^{K} \hat{x}_{k,t,n}.$$

For each neuron n, define the residual and total sums of squares

$$\mathrm{SS}_n^{\mathrm{res}} \;=\; \sum_{t=1}^{T} \big(\bar{x}_{t,n} - \hat{\bar{x}}_{t,n}\big)^2, \qquad \mathrm{SS}_n^{\mathrm{tot}} \;=\; \sum_{t=1}^{T} \big(\bar{x}_{t,n} - \bar{\bar{x}}_n\big)^2,$$

where $\bar{\bar{x}}_n \;=\; \frac{1}{T}\sum_t \bar{x}_{t,n}$ is the time-averaged PSTH for neuron n. The per-neuron coefficient of determination is

$$R_n^2 \;=\; 1 \;-\; \frac{\mathrm{SS}_n^{\mathrm{res}}}{\mathrm{SS}_n^{\mathrm{tot}}}.$$

The reported PSTH R^2 is the uniform-weighted average across neurons,

$$R^2_{\mathrm{PSTH}} \;=\; \frac{1}{N}\sum_{n=1}^{N} R_n^2$$

D.2 PSEUDO R^2 (POISSON)

For each neuron n, let $s_{i,n}$ and $\hat{s}_{i,n}$ denote the true spike count and predicted rate at sample $i \in \{1,\dots,M\}$ (where $M = K \cdot T$ after flattening), and let $\bar{s}_n = \frac{1}{M}\sum_i y_{i,n}$ be the per-neuron mean. Define the Poisson-style log-likelihood terms:

$$\ell_n^{\mathrm{model}} \;=\; \sum_{i=1}^{M} \Big[\, s_{i,n} \log\big(\hat{s}_{i,n}\big) \;-\; \hat{s}_{i,n} \Big]$$

$$\ell_n^{\mathrm{sat}} \;=\; \sum_{i=1}^{M} \Big[\, s_{i,n} \log\big(s_{i,n}\big) \;-\; \hat{s}_{i,n} \Big]$$

$$\ell_n^{\text{null}} \;=\; \sum_{i=1}^{M} \Big[\, s_{i,n} \log\big(\bar{s}_n\big) \;-\; \bar{s}_n \Big]$$

The per-neuron pseudo R^2 is

$$R^2_{\text{pseudo},\,n} \;=\; 1 - \frac{\ell_n^{\text{model}} - \ell_n^{\text{sat}}}{\ell_n^{\text{null}} - \ell_n^{\text{sat}}}$$

The reported population-level pseudo R^2 is the median across neurons,

$$R^2_{\text{pseudo}} \;=\; \underset{n \in \{1,\ldots,N\}}{\text{median}}\; R^2_{\text{pseudo},\,n}$$

Compared to the standard R^2, this measure benchmarks the model against the constant-rate null model under a Poisson noise assumption, and is more appropriate for spike-count targets.

E Supplementary Results and Figures

E.1 Scaling EI-NODE to larger and more brain regions

To test model scaling, we benchmarked 27 architectures on a NVIDIA GeForce RTX 4080. This sweep covered different population counts, latent dimensions per population, and population-MLP widths. Each architecture was measured for 20 epochs using one seed with our synthetic dataset and batch geometry. The first epoch was excluded from steady-state timing. The smallest configuration (3 populations, 1 latent/population, width 8) had 466,675 total parameters, 912 dynamics parameters, 1.847 s/epoch, 135.4 examples/s, and 427.0 MB peak allocated GPU memory. The largest configuration (12 populations, 4 latents/population, width 32) had 707,164 total parameters, 194,556 dynamics parameters, 59.479 s/epoch, 4.3 examples/s, and 784.1 MB peak allocated GPU memory. From the smallest to largest architecture, the capacity of the dynamics models increased 213.3x, steady-state epoch time increased 32.2x, and peak allocated memory increased 1.8x. These results are summarized in Table 1.

Table 1: Model configuration and performance across population and latent sizes

Populations	Latents/pop	Total latents	MLP width	Interaction MLPs	Total params	Dynamics params	Sec/epoch	Examples/sec	GPU memory (MB)
3	1	3	8	9	466,675	912	1.847	135.4	427.0
3	1	3	16	9	468,691	2,928	1.855	134.6	427.0
3	1	3	32	9	476,179	10,416	1.887	132.2	427.1
3	2	6	8	9	469,975	1,089	3.611	69.9	426.1
3	2	6	16	9	472,135	3,249	3.619	69.7	426.1
3	2	6	32	9	479,911	11,025	3.627	69.4	426.2
3	4	12	8	9	476,575	1,443	3.745	67.4	426.2
3	4	12	16	9	479,023	3,891	3.769	66.8	431.7
3	4	12	32	9	487,375	12,243	3.758	67.2	426.3
6	1	6	8	36	472,456	3,570	13.999	18.2	426.5
6	1	6	16	36	480,520	11,634	14.064	18.1	444.1
6	1	6	32	36	510,472	41,586	14.054	18.1	493.8
6	2	12	8	36	479,362	4,230	14.014	18.2	426.6
6	2	12	16	36	488,002	12,870	14.082	18.1	444.4
6	2	12	32	36	519,106	43,974	14.076	18.1	494.1
6	4	24	8	36	493,174	5,550	14.530	17.5	427.5
6	4	24	16	36	502,966	15,342	14.566	17.5	444.8
6	4	24	32	36	536,374	48,750	14.605	17.5	494.8
12	1	12	8	144	489,256	14,124	57.234	4.5	512.9
12	1	12	16	144	521,512	46,380	57.535	4.4	583.7
12	1	12	32	144	641,320	166,188	57.615	4.4	782.0
12	2	24	8	144	504,292	16,668	57.268	4.5	513.8
12	2	24	16	144	538,852	51,228	57.475	4.5	584.1
12	2	24	32	144	663,268	175,644	57.619	4.4	783.0
12	4	48	8	144	534,364	21,756	59.233	4.3	514.9
12	4	48	16	144	573,532	60,924	59.369	4.3	585.8
12	4	48	32	144	707,164	194,556	59.479	4.3	784.1

E.2 Simulated Decision-Making Task

E.2.1 Recovering Known Interactions

The EI-NODE model closely captures the ground-truth influences. Within the synthetic dataset, we extracted the synaptic inputs (currents) from each source population onto each target population. These currents serve as the model's ground-truth influences of one population onto another, and we tracked how each population thus influences one another over the time course of the experiment (a

'ground truth' for our estimated Fig. 3B). Visually, the input currents are indeed extremely similar to the EI-NODE's modeled influences (Fig. 3B vs Fig. 1). To quantitatively compare the estimated influences from the model to the ground-truth, we calculated the Pearson correlation coefficient (r) for each population pair (including self-connections) across all trials and summarized the results in Table 2.

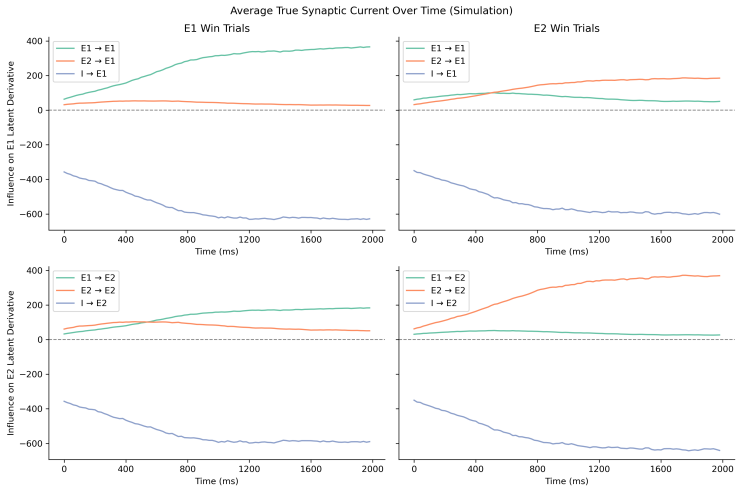

Figure 1: *Ground Truth Influences of Simulated Decision-Making Task*. Each panel shows the average synaptic current over time for each trial condition. The left (right) column shows the "E1 win" ("E2 win") trial condition. The top (bottom) row shows the average synaptic input into the E1 (E2) population. Green shows the average E1 synaptic current, orange shows the average E2 synaptic current, and blue shows the average I synaptic current.

Table 2: Pearson correlation coefficients between predicted and true influences by target-source pair.

Target	Source	r
E1	E1	0.96
E1	E2	0.87
E1	I	0.84
E2	E1	0.88
E2	E2	0.97
E2	I	0.85
I	E1	0.98
I	E2	0.98
I	I	0.90

E.2.2 EI-NODE Noise Robustness

To test the model's robustness to noise, we added noise to the simulated decision-making task in two different manners: 1) to the dynamics by increasing the level of external background Poisson drive to the neural populations, and 2) to the observations by resampling the original spiking activity from a Poisson distribution scaled by an independent random multiplicative gain.

In the first scenario, each neuron receives a random synaptic drive according to a Poisson noise model. Changing the synaptic drive alters the decision dynamics of the simulation, so the fixed-point structure is no longer observed within the true simulated spiking activity. Therefore, we only evaluated our model's ability to reconstruct the spiking activity. We increased the synaptic drive to 4x (medium SNR) and 16x (low SNR) the levels of the baseline simulation, and the EI-NODE model continues to accurately reconstruct the spiking activity. In the second scenario, we simulated noisy observations instead of noisy dynamics. For each neuron and trial within the baseline simulated dataset, we sample a random multiplicative gain, $g_{n,tr}$, from a Gamma distribution with mean 1

and variance $1/k$. We then resample the spiking activity for a given neuron and trial according to a Poisson distribution with a rate equal to $s_t * g_{n,tr}$, where s_t is the spike count for a single time bin t. We use a value of $k = 4$ (Coefficient of Variation = 0.5) and $k = 1$ (Coefficient of Variation = 1.0) to simulate observations of low and medium SNR, respectively. Under both conditions, the EI-NODE model continues to accurately infer the ground-truth influences and extract the same fixed-point structure as seen in the baseline simulation.

In both settings, we see that the EI-NODE model can still produce good predictions in the presence of noise (Fig. 2).

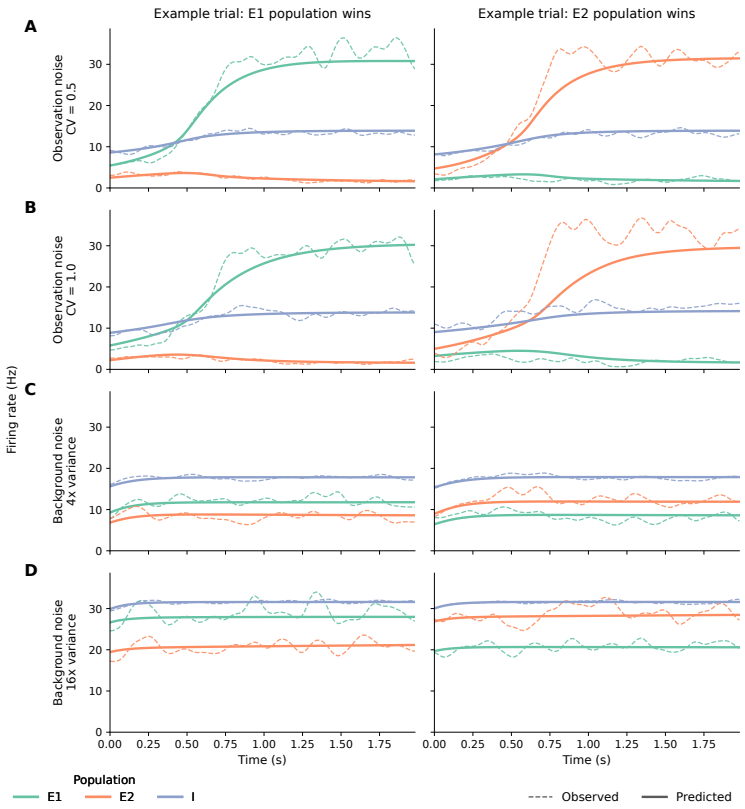

Figure 2: *Single-trial predictions of each noise scenario*. Comparison between population-averaged model predictions and true rates for example trials from each condition ("E1-wins" vs "E2-wins"). True rates were calculated by smoothing spiking counts with a 40ms gaussian kernel. Dashed lines show the true rates and solid lines show the predicted rates. Green, orange, and blue lines show the activity of E1, E2, and I populations, respectively. (A-B) Predictions for noisy observation environments for medium and low SNR trials, respectively. (C-D) Predictions for noisy dynamics environments for medium and low SNR trials, respectively.

E.2.3 TABLE OF R^2 SCORES

Table 3: Performance of dynamics models on the simulated decision-making dataset. Values are mean ± standard error (n=5). These metrics are computed across neurons from all populations.

Model	Avg. R^2	Pseudo R^2	PSTH R^2
EI-NODE	0.850 ± 0.002	0.255 ± 0.005	0.832 ± 0.001
Linear EI-NODE	0.809 ± 0.003	0.243 ± 0.004	0.804 ± 0.001
RNN	0.825 ± 0.002	0.252 ± 0.004	0.799 ± 0.002
EI-RNN	0.763 ± 0.007	0.242 ± 0.004	0.747 ± 0.003
NODE	0.859 ± 0.002	0.255 ± 0.004	0.831 ± 0.002
Linear NODE	0.827 ± 0.002	0.251 ± 0.004	0.813 ± 0.002

Table 4: Per-population performance of dynamics models on the simulated decision-making dataset, broken down by E1, E2, and I cell classes. Values are mean ± standard error (n=5).

	E1			**E2**			**I**		
Model	Avg. R^2	Pseudo R^2	PSTH R^2	Avg. R^2	Pseudo R^2	PSTH R^2	Avg. R^2	Pseudo R^2	PSTH R^2
EI-NODE	0.926 ± 0.003	0.271 ± 0.005	0.989 ± 0.001	0.927 ± 0.002	0.275 ± 0.004	0.988 ± 0.001	0.845 ± 0.002	0.007 ± 0.000	0.644 ± 0.003
Linear EI-NODE	0.909 ± 0.004	0.261 ± 0.005	0.983 ± 0.003	0.902 ± 0.003	0.261 ± 0.004	0.975 ± 0.003	0.777 ± 0.004	0.006 ± 0.000	0.594 ± 0.003
RNN	0.920 ± 0.002	0.268 ± 0.006	0.989 ± 0.001	0.917 ± 0.002	0.271 ± 0.004	0.988 ± 0.001	0.737 ± 0.007	0.006 ± 0.000	0.573 ± 0.005
EI-RNN	0.867 ± 0.004	0.257 ± 0.004	0.944 ± 0.003	0.907 ± 0.000	0.264 ± 0.004	0.978 ± 0.001	0.645 ± 0.009	0.005 ± 0.000	0.490 ± 0.006
NODE	0.933 ± 0.002	0.276 ± 0.005	0.991 ± 0.000	0.936 ± 0.002	0.273 ± 0.005	0.991 ± 0.001	0.848 ± 0.002	0.007 ± 0.000	0.638 ± 0.004
Linear NODE	0.919 ± 0.003	0.269 ± 0.005	0.989 ± 0.001	0.919 ± 0.002	0.268 ± 0.005	0.990 ± 0.001	0.772 ± 0.006	0.006 ± 0.000	0.602 ± 0.004

E.2.4 EI-RNN INFLUENCES

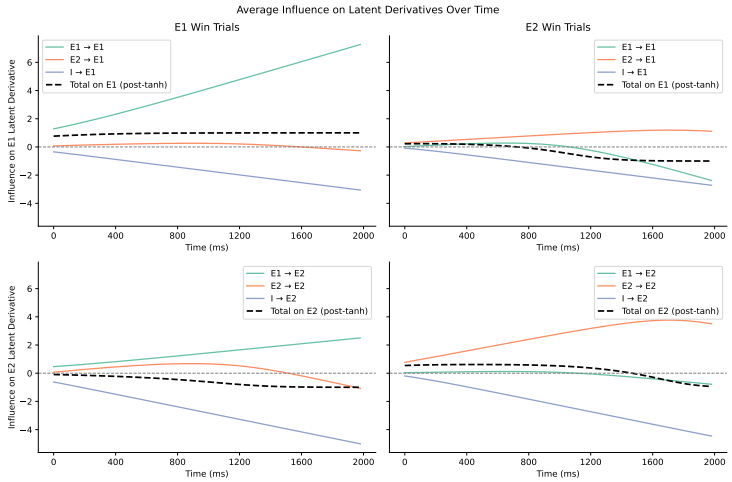

Figure 3: *EI-RNN Influences on the Simulated Decision-Making Task.* Average influences of the Dale's Law constrained RNN dynamics model for each condition of the simulated decision-making task. (Left Column) E1 win trials. (Right Column) E2 win trials. (Top row) Influence of each population on the latent derivative of population E1. (Bottom row) Influence of each population on the latent derivative of population E2. Note, the individual population influences shown do not sum to the true total influence (dashed black line) due to the $\tanh$ activation in the RNN definition (12). The total influence on each excitatory population for both conditions fails to approach zero at the end of the trials, showing that the EI-RNN model is incapable of finding fixed points.

E.2.5 VECTOR FIELDS OF MODEL COMPARISONS

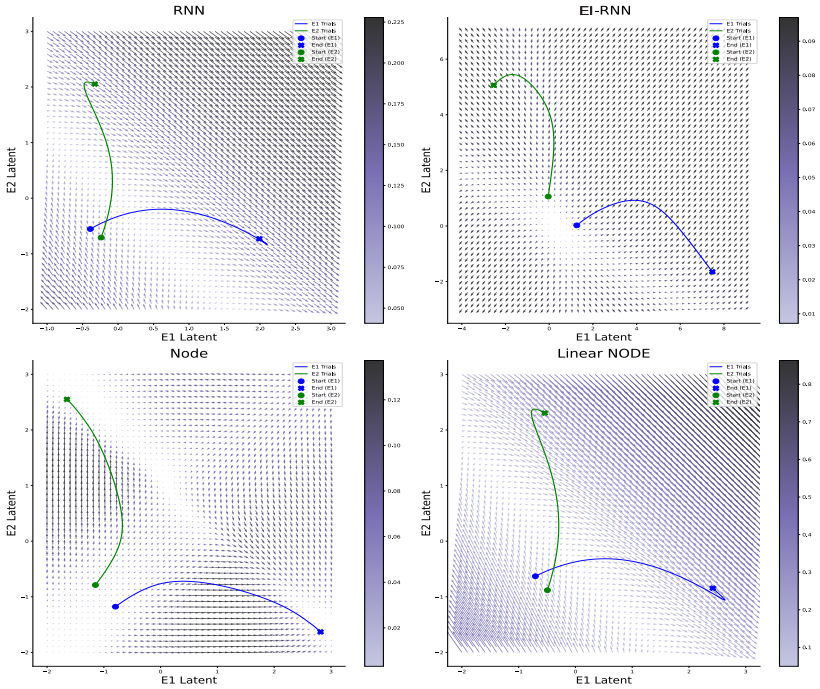

Figure 4: *Flowfields at 200 ms into the trial for comparison models*. (Top Left) RNN dynamics, (Top Right) Dale's Law constrained RNN Dynamics, (Bottom Left) NODE Dynamics, (Bottom Right) Linear NODE Dynamics.

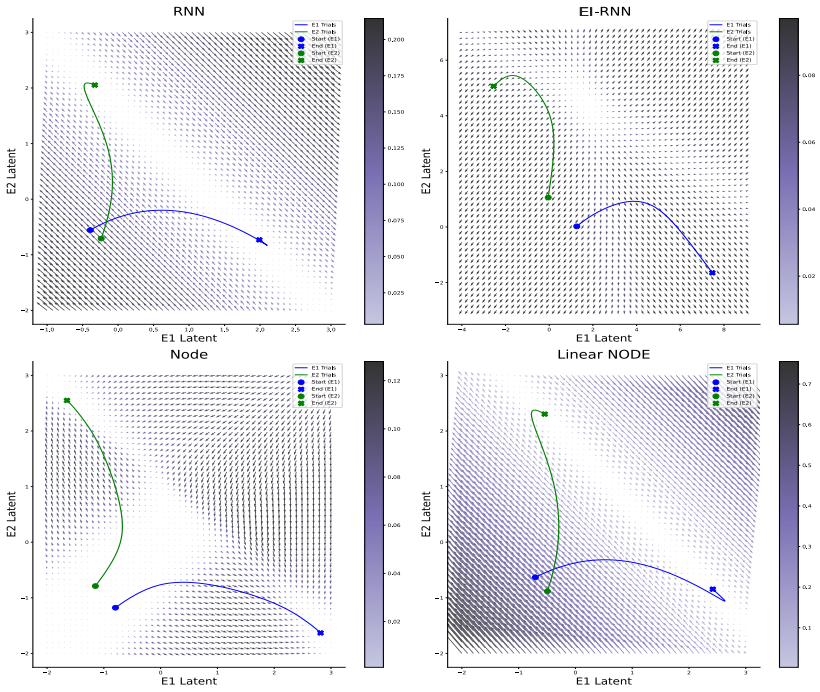

Figure 5: *Flowfields at 1800 ms into the trial for comparison models*. Only the NODE model is able to capture fixed points at the end of the trial, signifying the importance of both expressivity and nonlinearity. (Top Left) RNN dynamics, (Top Right) Dale's Law constrained RNN Dynamics, (Bottom Left) NODE Dynamics, (Bottom Right) Linear NODE Dynamics.

E.3 Table of Performance on Climbing Data

Table 5: Multisession climbing and perturbation pseudo-R^2. Each score is the mean across the two sessions. Training seeds are averaged within each dataset seed, and the reported value is the mean of those three split means. Brackets are Student-t 95% confidence intervals across the three split means. EI-RNN uses a linear input map; EI-RNN Input MLP uses a bias-free 64×64 input MLP.

Model	Climbing	Perturbation
EI-NODE	0.051 [0.043, 0.060]	0.157 [0.125, 0.189]
Linear EI-NODE	0.050 [0.037, 0.063]	0.142 [0.104, 0.180]
EI-RNN	0.049 [0.037, 0.062]	0.090 [0.056, 0.124]
Linear EI-RNN Input MLP	0.043 [0.010, 0.077]	0.074 [0.048, 0.100]
Linear EI-RNN Linear W_{ih}	0.038 [0.024, 0.053]	0.088 [0.061, 0.116]

E.4 Comparing Ground-Truth Currents to Learned Influences

E.4.1 Fitting CURBD to Climbing Data

To test our method's ability to recover ground-truth influences, we fit a two-area recurrent network (CURBD) to simultaneously recorded activity in primary and secondary motor cortex (M1 and M2) during the naturalistic climbing task (Perich et al., 2026). Spike counts were binned at 20 ms and smoothed with a causal Gaussian kernel ($\sigma = 1.5$ bins), and z-scored over time. The network used a hyperbolic-tangent nonlinearity, a membrane time constant of 200 ms, and an integration step of 4 ms. A small filtered white-noise input (amplitude 0.001, correlation time 100 ms) was present during fitting. Weights were learned with FORCE so that the network rates matched the z-scored target activity. Initial recurrent weights were drawn with a gain of 1.5 within and between areas, with a small negative offset (-0.1) on within-area weights.

To learn a network with Dale constraints, each presynaptic unit was constrained to a single sign. Its outgoing weights were parametrized to be non-negative, then multiplied by a learned scalar, so the unit was either excitatory or inhibitory. Inhibitory units made no synapses onto the other area, and excitatory synapses between areas were constrained to be non-negative. Within-area weights were left dense. Half of the units in each area were initialized as excitatory by ranking the mean within-area weight of each unit. After every training pass, identities could be updated, with each area held to at least 40% excitatory units.

Inter-region sparsity was applied to excitatory projections between areas. M1-to-M2 and M2-to-M1 projections were pruned separately. After each training pass, the excitatory source units with the weakest total cross-area weight were removed. The fraction removed increased from 40% to 80% across training. Within-area weights were not pruned. Inhibitory cross-area weights were held at zero for the whole fit, so the pruning removed only excitatory cross-area sources.

CURBD rates were mapped back through their z-score scalers and drawn as independent Poisson counts. Climbing trials were restricted to 60–260 ms of each bout to allow time for dynamics to develop. To generate pseudo-optogenetic perturbation trials, we targeted the stimulated region with a positive current for 25ms starting at the beginning of the trial, and we rolled out 200ms using the frozen learned weights. We used a current injection of 1 A, which was chosen to elicit a subsequent dip in activity in the other populations.

E.4.2 Fitting EI-NODE Models to CURBD Activity

We fit connectivity-constrained EI-NODE and EI-RNN variants to the spike counts from each CURBD model, using that model's excitatory and inhibitory labels. The four populations contained 26 (M1 E), 38 (M1 I), 50 (M2 E), and 75 (M2 I) units. Inputs were passed to the encoder and the dynamics as described in the Methods, with the unit pulse on inhibitory M1 at the first bin of each pseudo-optogenetic trial. We tested passing inputs to EI-RNNs via linear and nonlinear mappings (population MLP) to allow for an extensive comparison. The trials were split into 80% training, 10% validation, and 10% testing. All values are reported for the test set.

Models were scored based on their average of climbing and perturbation validation performance, and test performance is reported in Table 6. We found that EI-NODE performed better than EI-RNN on climbing (0.062 vs 0.030) and about the same on pseudo-opto (0.039 vs 0.043), while Linear EI-NODE performed best in pseudo-opto (0.062).

Table 6: Test pseudo-R^2 on CURBD spikes. Each row is the seed with the highest validation pseudo-R^2. The reported scores are the held-out test bouts.

Model	Input	Climbing	Pseudo-opto
EI-NODE	Input MLP	0.062	0.039
Linear EI-NODE	Input MLP	0.061	0.062
EI-RNN	Input MLP	0.030	0.043
EI RNN	Linear W_{ih}	0.028	0.030
Linear EI-RNN	Input MLP	0.029	0.011
Linear EI-RNN	Linear W_{ih}	0.011	0.007

E.4.3 Recovering Known Interactions

The influences from the best model fits (selected by best average performance across climbing and perturbation) for each architecture were then compared against the ground-truth currents used by the CURBD model used to generate activity. To assess the correlation between the learned influences and ground-truth currents, we plotted the Root-Mean-Squared (RMS) influence for each population for climbing, perturbation (without input), and perturbation (with input); see Fig.6. While the absolute magnitude of the RMS power differs between the fitted and generative models due to scaling differences in the architectures, the EI-NODE accurately captures the relative distribution of influences. Conversely, EI-RNN does not recover the ground-truth currents (finding incorrectly massive and unstructured influences), demonstrating that although the EI-RNN fits the simulated activity well, it did not recover the underlying circuit (Fig.7). We found that the nonlinear input helped EI-RNN model performance, but it was still below EI-NODE models and did not help influence correlation.

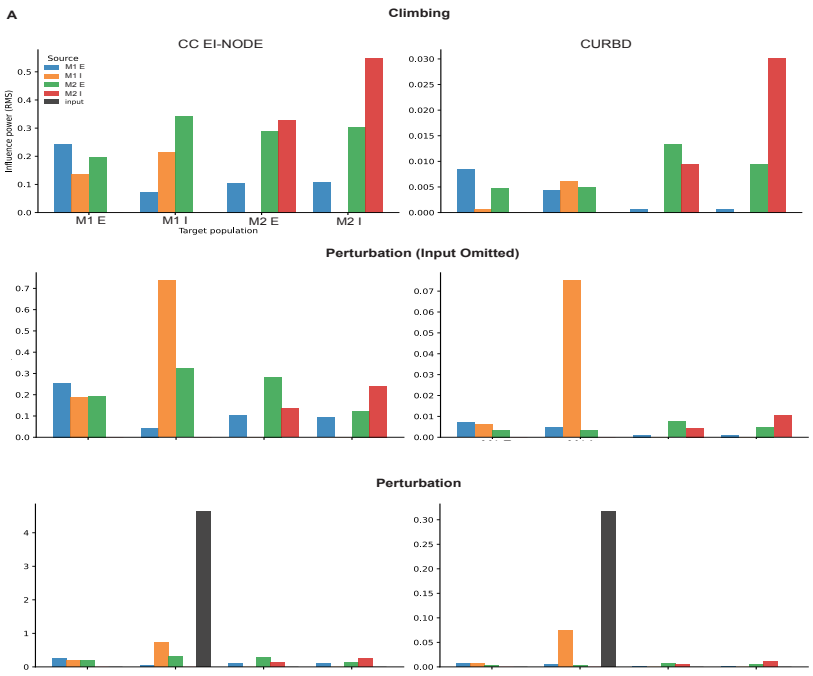

Figure 6: *EI-NODE recovers ground-truth CURBD influence currents.*Learned EI-NODE population influences. Bar height is the root-mean-square (RMS) of that influence over test trials, time, and target latent units. We compared RMS between the fitted EI-NODE model (left) and the ground-truth CURBD model (right) across three plotting conditions: climbing (top), perturbation with the external input influence omitted to reveal the scale of recurrent interactions (middle), and perturbation with the dominant external input included (bottom). The EI-NODE model successfully captures the relative distribution of the ground-truth currents within and across the primary (M1) and secondary (M2) motor cortex populations (the absolute magnitude scales differ between the models).

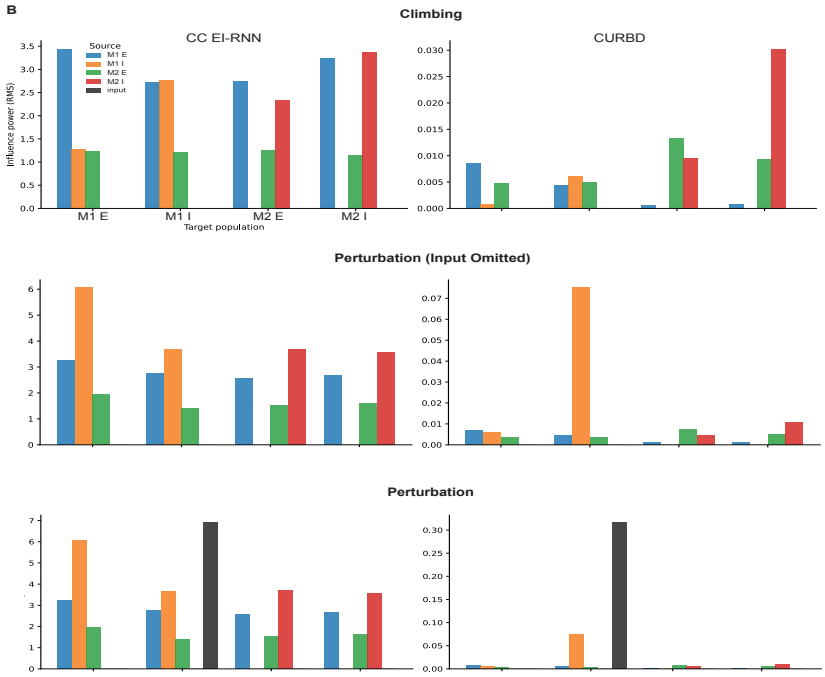

Figure 7: *EI-RNN fails to recover ground-truth CURBD influence currents.* Root-Mean-Squared (RMS) influence power is compared between the fitted EI-RNN model (left) and the ground-truth CURBD model (right) across three plotting conditions: climbing (top), perturbation with the external input influence omitted to reveal the scale of recurrent interactions (middle), and perturbation with the dominant external input included (bottom). Unlike the EI-NODE model, the EI-RNN influence distributions do not align with the underlying ground-truth circuit dynamics.