Introduction

Improvements in parallelization, automation, and cost per sample have led to an explosion of large datasets in biology, sociology, and other sciences. These advancements allow for the collection of detailed time series data in a variety of new fields. Often the goal of research is to use these data to learn about underlying generative processes. To do this in an interpretable manner, a mechanistic model is defined based on domain knowledge and parameterized based on the collected data.

There are two key obstacles to fitting these parameters. First: The measurable features of the system (muscle activity, stimulus state) at a given point in time do not completely contain the information for the generative process (e.g. neural firing rates). Moreover, these measurements generally are not generally experimentally accessible within the system. Second: The underlying process is non-linear. This precludes the use of any message-passing based algorithms analogous to hidden Markov models []. Together, this means there is no closed-form or standardized accessible solution to the inverse problem of parameter fitting for many relevant observable processes.

Therefore, we designed a new, neural network based method called Parameter Estimates from Observables (ParEO). This method takes a time series of observables (X_t) and uses them to infer the corresponding series of generative latent states (Q_t) with a learned function η. Similarly, the observables are generated from the latent state through a measurement function $\mathcal{M}$. The latent states evolve according to a manually defined mechanistic model Θ with parameters π. The goal of this method is to estimate π for a given condition.

Here, we show that ParEO accurately parameterizes both linear and complex non-linear mechanistic models from incomplete data.

Model Architecture:

In this methodology, we assume access to experimental time course data X_t which is the measurable value generated by some latent process with states Q_t.

We first define the measurement function $\mathcal{M}(Q_t) = X_t$. In most cases, we have a reasonable expectation of the form of $\mathcal{M}$, with the simplest form being a projection of some component(s) of the latent state. However, there is no logical barrier in the ParEO method to $\mathcal{M}$ taking a more complicated or non-linear form with trainable weights, as long as it contains information on the latent state.

Next, we address the inference function η. In general, we cannot assume that the measurement process is invertible, and a single timepoint of the observable X_t cannot be expected to contain sufficient information to reconstruct the latent state. However, by Taken's theorem, we *can* state that a lagged embedding of the observable $\tilde{X}_t = \{X_t, X_{t-\tau}, \dots, X_{t-m\tau}\}$ contains sufficient information to describe the dynamics of our d dimensional latent attractor if $m \geq 2d + 1$ []. Since we can state that there is some transformation from the embedded observable to the latent space and we are agnostic to the form of the functional transformation, we can define the inference process $\eta(\widetilde{X_t}) = Q_t$ using a neural network. In principle, this can always be done with a sufficiently complex deep neural network, though in practice we have found that using a recurrent architecture significantly improves performance with limited data.

Finally, we formalize the mechanistic model Θ. In general, this function should be a system of differential equations describing the latent state interactions. The set of all free parameters in the model are defined by π. The only requirement of the function is that it can be written in Tensorflow such that $\Theta(Q_t) = Q_{t+1}$ and recursively called such that $\Theta^\gamma(Q_t) = Q_{t+\gamma}$. Beyond that, the benefit of the entire ParEO process is being able to freely define this model based on interpretability and prior knowledge of the system.

Training:

We define three loss functions to fit the parameters of the ParEO network. The first is a reconstruction loss:

$$\mathcal{L}_{recon.} = \sum_{t=0}^{T} \left(X_t - \mathcal{M}\left(\eta\left(\tilde{X}_t\right)\right)\right)^2.$$

This term captures how well the observable is preserved through the inference and measurement networks. Second is the latent prediction loss:

$$\mathcal{L}_{latent\ pred.} = \sum_{t=0}^{T} \left(\Theta^\gamma\left(\eta\left(\tilde{X}_t\right)\right) - \eta\left(\tilde{X}_{t+\gamma}\right)\right)^2 = \sum_{t=0}^{T} \left(\Theta^\gamma(Q_t) - Q_{t+\gamma}\right)^2.$$

This term captures the accuracy of the time evolution given the current inferred latent states. The third is the prediction loss:

$$\mathcal{L}_{pred} = \sum_{t=0}^{T} \left(\mathcal{M}\left(\Theta^\gamma\left(\eta\left(\tilde{X}_t\right)\right)\right) - X_{t+\gamma}\right)^2.$$

These losses are weighted by hyperparameters β_i to give a cumulative loss function:

$$\mathcal{L} = \beta_1 \mathcal{L}_{recon.} + \beta_2 \mathcal{L}_{latent\ pred.} + \beta_3 \mathcal{L}_{pred}$$

This loss can now be used to simultaneously train the entire network. We have found that this combination of losses is sufficient to prevent a collapse of the latent inference to trivial solutions (e.g. all latent states map to 0) and achieve latent parameter estimates π with high correlation to the true value. Furthermore, we have found that $\mathcal{L}_{latent\ pred.}$ And $\mathcal{L}_{lpred.}$ are largely sufficient to constrain the optimization. Accordingly, weights of $\beta_1 = 1,\ \beta_2 = 30, \beta_3 = 30$ are used throughout this work. We present several test cases and variations to ParEO below.

Spring model:

We first show that ParEO can correctly solve for latent parameters in a simple linear system—a mass on a spring. The latent dynamics (and therefore Θ) are given by:

$$\dot{y} = v$$

$$\dot{v} = -\left(\frac{k}{m}\right) y$$

$$Q_t = \{y_t, v_t\}, \qquad \pi \equiv \left\{\frac{k}{m}\right\}$$

To illustrate the use of ParEO and provide intuition, we consider the case where our measurement is only able to observe the position of the mass such that $\mathcal{M}(Q_t) = y_t$. It is clear to see in this situation that while the full latent state (position, velocity) cannot be estimated from a single timepoint measurement, it can be easily inferred from the embedded vector $\tilde{X}_t$. For the sake of demonstration, we will still use a recurrent neural network for the inference network η. To test performance, we simulate the system under a wide dynamic range of parameter conditions. For each case, we then fit the full architecture and attain an estimate for the latent parameter π (Fig. 2b). We can easily see that the parameters are in good agreement across multiple orders of magnitude. Additionally, the breakdown of the system at high parameter values is due to information limitation and Nyquist's theorem (red line shows where period=measurement frequency), and the failure at low values is due to the decreasing signal to noise ratio.

In the process of attaining parameter estimates, ParEO automatically