Expert Controlled Neural Differential Equations: Disease Progression Modelling

Yunjie Tim Wang
Department of Statistics
University of Oxford
`yunwang@stats.ox.ac.uk`

Abstract

Incorporating expert knowledge into machine learning models has succeeded in various scientific domains. In particular, in pharmacology, the Latent Hybridisation Model (LHM) has exhibited remarkable predictive performance on real-world intensive care data of COVID-19 patients. However, its assumption about the expert model being self-contained might be restrictive in certain scenarios. To remedy this, we propose a principled method using ideas from neural controlled differential equations, which enjoy well-developed mathematical foundations. We provide an alternative way to leverage expert knowledge and demonstrate competitive predictive performance of our model on simulated datasets.

1 Introduction

Accurately modelling and predicting patient health status based on clinical observations and treatment plans is paramount in healthcare. To address this challenge, the Latent Hybridisation Model (LHM) (Qian et al., 2021) attempts to combine mechanistic models with machine learning approaches; it assumes that the evolution of a patient's physiological status can be jointly described by a mechanistic model formed by human experts and a neural Ordinary Differential Equation (neural ODE) (Chen et al., 2019). The main difficulty of this task lies in modelling the interaction of the expert variables with those additional variables, as those variables may not have any meaningful interpretations to be incorporated as part of the hand-crafted ODE system. This interaction is often left unmodelled (Yazdani et al., 2020; Xu and Valocchi, 2015) or only one-way interaction is included (Qian et al., 2021) (See Figure 1 for an illustration).

This leads us to consider an alternative approach that uses the expressiveness of machine learning methods to learn this interaction from data, while including information from the expert models. However, the usual neural ODE cannot update its dynamics according to another system, as once the parameters are learnt, the trajectory only depends on the initial value.

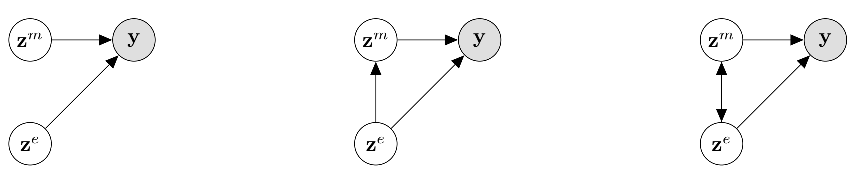

Figure 1: Graphical models for an unmodelled interaction (left), the one-way interaction in LHM, and a desired model (right). $\mathbf{y}$ is the output observation, $\mathbf{z}^e$ denotes the expert variable, and $\mathbf{z}^m$ stands for the additional variables.[1]

[1] We abuse the notation from (Evans and Richardson, 2013) to represent the dependence using double arrows.

Preprint. Under review.

1.1 Neural Controlled Differential Equations

Fortunately, the neural controlled differential equations (NCDEs) (Kidger et al., 2020) resolve this issue for us. NCDEs incorporate incoming information by computing a Riemann-Stieltjes integral with respect to the external dynamics $X(t) : [\tau, T] \to \mathbb{R}^v$, which is a continuous function of bounded variation. We define the hidden dynamics by:

$$\mathbf{z}(t) = \mathbf{z}(\tau) + \int_{\tau}^{t} \mathbf{f}(\mathbf{z}(s); \boldsymbol{\theta}) \mathrm{d}X(s) \tag{1}$$

where $\mathbf{f}$ is a vector field parameterised by a neural network and the initial value $\mathbf{z}(\tau)$ is also learned by a network.

Usually the function $X(s)$ is an interpolation of irregular observations, but here we can replace the $\mathrm{d}X(s)$ with the known expert equations. Nonetheless, the NCDE requires a valid initial value of the latent variables and the information from the expert variables cannot *directly* contribute to the output (the arrow $\mathbf{z}^e \to \mathbf{y}$ is missing, see Figure 1).

1.2 Contribution

We demonstrate how to use NCDEs for integrating the expert ODEs by careful construction of the function $\mathbf{f}$ governing the hidden dynamics. Theoretical results are provided to justify our construction. In addition, we apply our models to prediction tasks on simulated datasets, where we achieve state-of-the-art performance.

1.3 Related Works

We build our work on (Qian et al., 2021) and apply our model to the prediction tasks in the same work. The application of NCDEs in healthcare has also appeared in several other works; Seedat et al. (2022) propose the treat-effect neural controlled differential equation (TE-CDE) for counterfactual outcome prediction, while Hess et al. (2023) build on TE-CDE by incorporating more principled uncertainty estimation using a Bayesian approach. However, those works do not consider a hybrid between expert models and data-driven methods, which is our main focus here.

2 Problem Formulation

We consider the same problem set up as in (Qian et al., 2021). Given a set of N patients indexed by $i \in \{1, 2, \ldots, N\}$, each patient i is associated with a set of n_i observations of a vector-valued function $\mathbf{y}_i(t)$ describing their health status and a trajectory of another vector-valued function $\mathbf{a}_i(s)$ representing the treatment plan for the patient. Namely, the observations $\mathcal{D}_i$ for patient i is:

$$\mathcal{D}_i = \{\mathbf{y}_i(t)\}_{t \in \mathcal{T}_i} \cup \{\mathbf{a}_i(s)\}_{s \in [0,T]} \tag{2}$$

where $\mathcal{T}_i := \{t_{i_1}, \ldots, t_{i_{n_i}}\} \subset [0, T]$ is a discrete set within the entire time horizon $[0, T]$ of time-points at which those data are recorded. The function $\mathbf{y}_i(t)$ usually consists of measurements of physiological variables.

We are interested in the future measurements $\mathbf{y}_i(t)$ of the patient given this individual's past measurements up to a time t_0 and the treatment trajectory $\{\mathbf{a}_i(s) \mid s \in [0, T]\}$ over the time horizon. That is we would like to predict the future physiological measurements given the available information:

$$\Pr(\{\mathbf{y}_i(t)\}_{t \in [t_0, T]} \mid \{\mathbf{y}_i(t)\}_{t \in \mathcal{T}_i \cap [0, t_0]} \cup \{\mathbf{a}_i(s)\}_{s \in [0,T]}\}) \tag{3}$$

Various challenges need to be resolved for accomplishing this task. First, the observations from the measurements taken from patients $\{\mathbf{y}_i(t)\}_{t \in \mathcal{T}_i \cap [0, t_0]}$ are not necessarily evenly spaced in time, which is common in a clinical context; this renders many of the traditional regular time-series approaches inapplicable. Second, due to this problem setting, there may be a scarcity of data, which limits the performance of some of the deep learning based models.

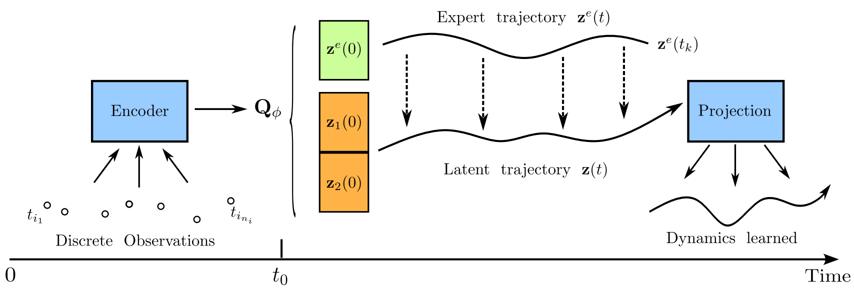

Figure 2: The VAE architecture of our expert CDEs. The dotted arrows illustrate the control of expert variables exerted on the latent dynamics. The values from expert $\mathbf{z}^e(t)$ can be incorporated either implicitly (the neural CDE (6)) or explicitly (our model (7)).

3 Methodology

3.1 Motivation

To address those challenges, we follow closely the latent hybridisation model (LHM) approach proposed in (Qian et al., 2021). Traditional mechanistic models in pharmacology use a set of expert variables $\mathbf{z}^e(t) \in \mathbb{R}^E$ to model the evolution of the physiological status of patients under treatment (Katzung, 2012). However, those variables may not be directly observable in a clinical setting, and they may not completely model the dynamics given the complexity of the system. This motivates us to combine the expert model with data-driven machine learning methods.

We make the standard assumption that the observed irregular clinical measurements $\mathbf{y}_i(t)$ are governed by a latent dynamical system. Part of this dynamics is determined by the expert model:

$$\frac{\mathrm{d}\mathbf{z}^e(t)}{\mathrm{d}t} = \mathbf{f}^e(\mathbf{z}^e(t), \mathbf{a}(t); \boldsymbol{\theta}^e) \tag{4}$$

where the function $\mathbf{f}^e : \mathbb{R}^{E\times A} \to \mathbb{R}^E$ with parameters $\boldsymbol{\theta}^e$, which are calibrated through experiments. To model the full dynamics through data-driven methods, Qian et al. (2021) directly incorporate the expert ODE (4) into a neural ODE (Chen et al., 2019),

$$\frac{\mathrm{d}\mathbf{z}^m(t)}{\mathrm{d}t} = \mathbf{f}^m(\mathbf{z}^m(t), \mathbf{z}^e(t), \mathbf{a}(t); \boldsymbol{\theta}^m) \tag{5}$$

where $\mathbf{f}^m : \mathbb{R}^{M\times E\times A} \to \mathbb{R}^E$ is a neural network with unknown weights $\boldsymbol{\theta}^m$ and $\mathbf{z}^m(t)$ are additional latent variables. Due to the assumption of a self-contained expert model (Qian et al., 2021), there is only explicit dependence of $\mathbf{z}^m(t)$'s dynamics on the expert variables but not vice versa. However, this assumption may not be always valid, especially in the context of modelling a complex system in science (Jaffe et al., 2023; Hu, 2014).

3.2 Expert Controlled Differential Equations

To this end, we propose an alternative approach that not only incorporates the expert information but also allows for modelling the implicit interaction among all latent variables.

We start by considering a single latent variable, $\mathbf{z}(t) \in \mathbb{R}^{M+E}$ modelling the dynamics of the entire latent space. It is governed by a neural CDE (Kidger et al., 2020) that is driven by the expert variables $\mathbf{z}^e(t)$. The initial values of the expert variables are jointly learned with $\mathbf{z}(0)$ through the VAE as in (Qian et al., 2021). The intuition is that the latent dynamics is constrained (controlled) by expert equations formed using domain-specific knowledge. We thus have the following equation for the latent dynamics:

$$\mathbf{z}(t) = \mathbf{z}(0) + \int_0^t \mathbf{h}(\mathbf{z}(s); \boldsymbol{\theta})\mathrm{d}\mathbf{z}^e(s) \tag{6}$$

where $\mathbf{h} : \mathbb{R}^{M+E} \to \mathbb{R}^{(M+E)\times E}$ is a matrix-valued neural network parameterised by $\boldsymbol{\theta}$ and $\mathrm{d}\mathbf{z}^e(s) = \frac{\mathrm{d}\mathbf{z}^e(s)}{\mathrm{d}s}\mathrm{d}s$ can be computed using Equation (4). The treatment plan is implicitly encoded in the expert equations as implemented in (Qian et al., 2021); we can also use another neural CDE to update the dynamics with the treatment plan (Hess et al., 2023).

Next, to further leverage information from the expert equations for prediction, we note that when $\mathbf{h}$ is the identity matrix, we recover the dynamics of the expert variables $\mathbf{z}^e(t)$. It is also easy to see that if the neural CDE is driven by a constant vector of ones (*i.e.* $\frac{\mathrm{d}\mathbf{z}^e(t)}{\mathrm{d}t} = \mathbf{1}$), it reduces to a neural ODE that does not rely on the expert equations. This motivates us to consider an ensemble-like approach (Dietterich, 2000; Yazdani et al., 2020; Xu and Valocchi, 2015) that interpolates between a mechanistic model and a fully data-driven model. Namely, we associate the first E coordinates in $\mathbf{z}(t)$ with expert variables and allow them to receive extra information from the expert dynamics:

$$\mathbf{z}(t) = \mathbf{z}(0) + \int_0^t \alpha \begin{bmatrix} \mathrm{I} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} (1-\alpha)J_{E\times E} \\ J_{M\times E} \end{bmatrix} \odot \mathbf{h}(\mathbf{z}(s);\boldsymbol{\theta})\mathrm{d}\mathbf{w}^e(s) \tag{7}$$

where $\mathrm{d}\mathbf{w}^e(s) = \beta\mathrm{d}\mathbf{z}^e(s) + (1-\beta)\mathrm{d}s\mathbf{1}$ is a weighted average of the expert dynamics and the vector encoding only information about the time[2]. Here $J_{n\times n}$ is a $n \times n$ matrix of ones and the symbol $\odot$ denotes the Hadmard (element-wise) product between two matrices. The $\alpha, \beta \in [0,1]$ can be fixed as hyperparameters; α can also be treated as a trainable parameter in the neural CDE.

Note that in contrast to (Qian et al., 2021), the dynamics of the latent space is evolved together using $\mathbf{h}$. In fact, we note that their LHM (5, 6) is a slight variation of Equation (7).

3.3 Theoretical Justification

By re-writing the solution to LHM (5, 6) with the notation $\mathbf{z}(t) = \begin{bmatrix} \mathbf{z}^e(t) \\ \mathbf{z}^m(t) \end{bmatrix}$, we have the following:

$$\mathbf{z}(t) = \mathbf{z}(0) + \int_0^t \begin{bmatrix} \mathbf{f}^e(\mathbf{z}^e(s)) \\ \mathbf{f}^m(\mathbf{z}^m(s), \mathbf{z}^e(s)) \end{bmatrix} \mathrm{d}s \tag{8}$$

$$= \mathbf{z}(0) + \int_0^t \begin{bmatrix} \mathrm{I} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g}(\mathbf{z}(s)) \end{bmatrix} \mathrm{d}\mathbf{z}^e(s) \tag{9}$$

which corresponds to the case where $\alpha = 1, \beta = 1$ with $\mathbf{g}$ being some neural network that may be parameterised to be arbitrarily expressive. Additionally, this partially coincides with a theoretical result stated in (Kidger et al., 2020).

Theorem 3.1. *(Informal) Given a continuous function of bounded variation X representing the observations, any equation of the form $\mathbf{z}_t = \mathbf{z}(0) + \int_0^t \mathbf{f}(\mathbf{z}(s), X(s))\mathrm{d}s$ can be represented by a Neural CDE of the form $\mathbf{z}_t = \mathbf{z}(0) + \int_0^t \mathbf{g}(\mathbf{z}(s))\mathrm{d}X(s)$.*

Combined with the universal approximation property of neural CDEs (Kidger, 2022), we see that our formulation is well-justified.

3.4 Practical Implementation

The implementation is similar to the VAE architecture in (Qian et al., 2021), where we first pass the discrete observations up to t_0 through an LSTM encoder to learn a representation by the variational distribution $\mathbf{Q}_\phi$ and then we evolve the latent dynamics according to Equation (7). The whole trajectory of the latent dynamics is then passed to a projection layer to recover the measurement values (Figure 2). The model is trained and evaluated in the same way as the LHM.

We also remark that it is possible to learn a separate VAE that only uses the expert equations (4) or the interpolation of observations (Kidger et al., 2020) to evolve the dynamics of $\mathbf{z}^e(t)$ in the latent space, and we can use this to drive the dynamics of the neural CDE. This additional flexibility of our model potentially allows for incorporating a wider range of expert knowledge into the model (*e.g.* discrete dynamical systems).

[2] Thus, the CDE is driven by altered observations

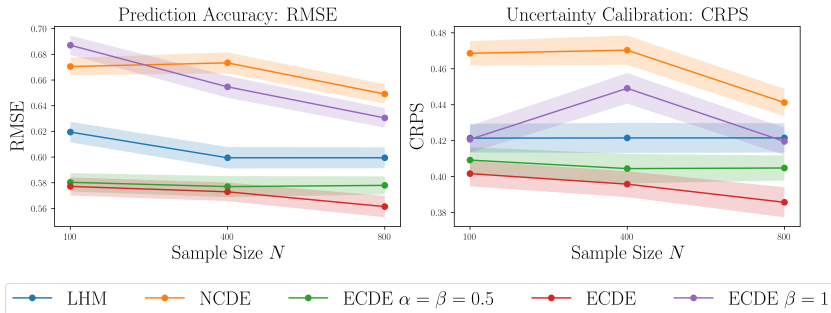

Figure 3: Comparison of the predictive performance of models trained with $N = 100, 400, 800$ samples and evaluated at $t_0 = 5$. The ECDE method scales better as more data become available.

4 Experiments

We train and evaluate our models using the simulated data provided in (Qian et al., 2021) with $D = 40$ observable physiological variable and the number of un-modelled states $M = 4$. We compare our models' predictive performance for a time horizon of $T = 14$ in terms of Root Mean Squared Error (RMSE) and Continuos Ranked Probability Score (CRPS) against the state of the art LHM model proposed by the same authors. For fair comparison of dynamics modelling, we use the same encoder structure for all models. The result is averaged over 50 iterations to account for uncertainty.

We report the performance of the neural controlled differential equation (NCDE) (6), the expert controlled differential equations (ECDE) (7) with a trainable α parameter and fixed some $\beta \sim \mathrm{Unif}(0, 1)$, as well as two variants of it: 1) removing the weighting on expert dynamics parameter (*i.e.* $\mathrm{d}\mathbf{w}^e(s) = \mathrm{d}\mathbf{z}^e(s)$ with $\beta = 1$) and 2) fixing $\alpha = \beta = 0.5$. In the neural CDE and the ECDE where $\beta = 1$, we include an extra time dimension as suggested by Kidger (2022). For the ECDE with $\beta > 0$, we observe that including an extra dimension of time does not alter its performance much.

Table 1: Comparison of predictive performance in terms of RMSE for $t_0 = 2, 5, 10$ trained using 400 samples.

Model	$t_0 = 2$	$t_0 = 5$	$t_0 = 10$
LHM	**0.551 (0.003)**	0.599 (0.004)	0.573 (0.005)
NCDE	0.567 (0.004)	0.640 (0.004)	0.586 (0.003)
ECDE	0.553 (0.003)	**0.573 (0.004)**	**0.572 (0.005)**
ECDE ($\alpha = \beta = 0.5$)	0.557 (0.003)	0.577 (0.004)	0.578 (0.006)
ECDE ($\beta = 1$)	0.580 (0.004)	0.600 (0.004)	0.591 (0.004)

Our results verify the effectiveness of including the parameters α and β in Equation (7). As shown in Table 1 and Figure 3, our proposed expert controlled differential equation (ECDE) has exhibited state of the art predictive performance on the simulation data, which aligns with the theoretical result we established in Section 3.3.

5 Limitations and Future Works

Choice of parameters α, β In our ECDE formulation (7), parameters α and β were introduced as weighting terms. Those were either chosen based on our prior belief about how well the expert model can perform or optimized during model training. Additionally, random initialisation with $\beta \sim \mathrm{Unif}(0, 1)$ has also proven to be beneficial for performance. More principled ways for finding those parameters can be explored in future works using ideas from ensemble learning or meta learning (Shu et al., 2019).

Uncertainty quantification Both LHM and ECDE use the VAE architecture due to the intractability of marginal likelihood. While this allows for uncertainty quantification, recently Miani et al. (2022)

have shown that uncertainty captured by VAEs may be inaccurate and unreliable, which is detrimental to clinical applications. Therefore, alternative fully Bayesian approaches may be considered (Xu et al., 2022; Hess et al., 2023). We briefly discuss how to combine our approach with the infinitely deep Bayesian neural network in Appendix A.1.

Real world scenarios While our model has exhibited good ability to scale with data, we only performed experiments on simulated datasets. It is left to future work to evaluate the ECDEs on real-world datasets that may have significantly more complex latent dynamics.

6 Conclusion

In this paper, we introduce the Expert-Controlled Differential Equation (ECDE) model for pharmacology and disease progression modelling. Our model is based on a novel modification of the usual neural controlled differential equation. The main strengths of the ECDE model is its ability to model the interaction within the latent space, while incorporating the expert information. Additionally, we provide theoretical evidence to support the model formulation and show that it achieves the state-of-the-art performance through empirical results.

A Appendix

A.1 Extension to Bayesian neural ODEs

We briefly mention how to incorporate uncertainty estimation into the current framework using infinitely deep Bayesian Neural Networks (BNN) (Xu et al., 2022). The infinitely deep BNN has two main components: a neural-network function governing the dynamics of hidden states $\mathbf{h}(\mathbf{z}(t), t; W_t)$ with time-varying weights W_t and an Ornstein-Uhlenbeck (OU) prior process governing the evolution of the weights, whose variational posterior is formulated as:

$$\mathrm{d}W_t = \mathbf{f}(W_t, t; \boldsymbol{\phi})\mathrm{d}t + \sigma \mathrm{d}B_t \tag{10}$$

where B_t is the Brownian motion, $\mathbf{f}$ is a small neural network with weights ϕ, and $\sigma > 0$ is the drift coefficient. To combine this with our framework, we can either proceed as in (Hess et al., 2023) to parameterise the neural CDE using the stochastic weights, or we can instead make the evolution of W_t depend on the observed expert dynamics:

$$W_t = W_0 + \int_0^t \mathbf{f}(W_s; \boldsymbol{\phi})\mathrm{d}\mathbf{z}^e(s) + \sigma B_t \tag{11}$$

In this way, the evolution of $\mathbf{z}(t)$ depends on the expert dynamics implicitly. However, this method has significantly more parameters than our original ECDE approach, which is in disadvantage when data is scarce.

A.2 Code and Implementation

We have implemented all our methods using the `torchdiffeq` library (Chen, 2018). We have also incorporated the infinitely deep BNN using the `torchsde` library (Kidger et al., 2021; Li et al., 2020) but we did not evaluate those models on the simulated datasets.

Our implementation can be found at the following GitHub repository, which is also based on (Qian et al., 2021).

References

Chen, R. T. Q. (2018). torchdiffeq.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2019). Neural ordinary differential equations.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, page 1–15, Berlin, Heidelberg. Springer-Verlag.

Evans, R. J. and Richardson, T. S. (2013). Marginal log-linear parameters for graphical markov models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 75(4):743–768.

Hess, K., Melnychuk, V., Frauen, D., and Feuerriegel, S. (2023). Bayesian neural controlled differential equations for treatment effect estimation.

Hu, C. (2014). Exposure–response modeling of clinical end points using latent variable indirect response models. *CPT: pharmacometrics & systems pharmacology*, 3(6):1–8.

Jaffe, P. I., Poldrack, R. A., Schafer, R. J., and Bissett, P. G. (2023). Modelling human behaviour in cognitive tasks with latent dynamical systems. *Nature Human Behaviour*, 7(6):986–1000.

Katzung, B. G. (2012). *Basic and clinical pharmacology*.

Kidger, P. (2022). On neural differential equations.

Kidger, P., Foster, J., Li, X., Oberhauser, H., and Lyons, T. (2021). Neural SDEs as Infinite-Dimensional GANs. *International Conference on Machine Learning*.

Kidger, P., Morrill, J., Foster, J., and Lyons, T. (2020). Neural controlled differential equations for irregular time series.

Li, X., Wong, T.-K. L., Chen, R. T. Q., and Duvenaud, D. (2020). Scalable gradients for stochastic differential equations. *International Conference on Artificial Intelligence and Statistics*.

Miani, M., Warburg, F., Moreno-Muñoz, P., Detlefsen, N. S., and Hauberg, S. (2022). Laplacian autoencoders for learning stochastic representations.

Qian, Z., Zame, W. R., Fleuren, L. M., Elbers, P., and van der Schaar, M. (2021). Integrating expert odes into neural odes: Pharmacology and disease progression.

Seedat, N., Imrie, F., Bellot, A., Qian, Z., and van der Schaar, M. (2022). Continuous-time modeling of counterfactual outcomes using neural controlled differential equations.

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. (2019). Meta-weight-net: Learning an explicit mapping for sample weighting.

Xu, T. and Valocchi, A. J. (2015). Data-driven methods to improve baseflow prediction of a regional groundwater model. *Computers & Geosciences*, 85:124–136.

Xu, W., Chen, R. T. Q., Li, X., and Duvenaud, D. (2022). Infinitely deep bayesian neural networks with stochastic differential equations.

Yazdani, A., Lu, L., Raissi, M., and Karniadakis, G. E. (2020). Systems biology informed deep learning for inferring parameters and hidden dynamics. *bioRxiv*.